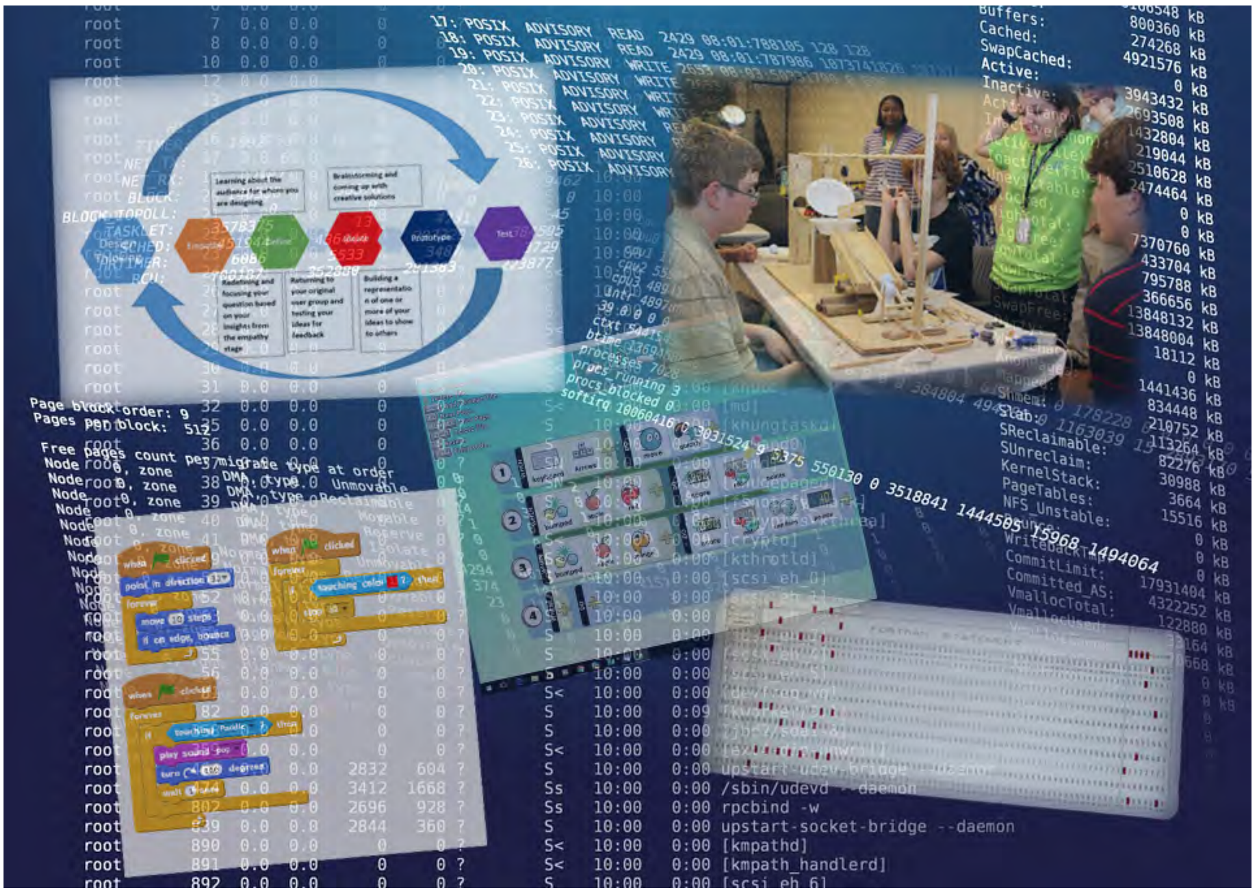
Education and Technology: Manitoba Makers and Coders

A Snapshot of Maker and Coder Ideas in Manitoba in 2017.



Edited by:
Reynold Redekopp, PhD (University of Manitoba)
Mike Nantais, PhD (Brandon University)

ISBN: 978-0-9958087-1-3

Acknowledgements

The editors wish to acknowledge the authors who submitted their work to this publication. They are all active, busy educators who felt it important to share their work with others. Thanks also go to the board of the Manitoba Association for Computing Educators (ManACE), who supported this venture and are helping to distribute this volume across the province. Several ManACE board members are authors in this volume as well.

Please don't blame the authors for any formatting peculiarities that you may find - they are our errors.

Mike Nantais originated the idea of putting together the first volume. We enjoyed the process so we sat down after a ManACE board meeting and began the process of identifying a relevant topic for volume 2 and decided to focus on the Maker and Coder movements. The response from the people we approached to write a chapter was overwhelmingly positive. We are so grateful to all those who agreed to participate - even the few whose work/family did not allow them to finish. We hope they will be in the next volume as will the many other fantastic teachers who we didn't ask this time; there are so many of them.

Cover images:
https://commons.wikimedia.org/wiki/File:Design_thinking.png
https://www.army.mil/article/41515/13_steps_to_pop_a_balloon
https://pixabay.com/en/code-program-source-computer-pc-113611/
https://en.wikipedia.org/wiki/Punched_card
Screenshots by R. Redekopp

About the Editors

Mike Nantais. PhD, is an Associate Professor at Brandon University. He is a teacher educator with an interest in educational technology, in particular, the intersection of education and social media. Mike came to the university after a 3 year stint as the Principal of a K-12 school and a 30 year career in public schools where he taught mathematics, sciences, and a variety of computer courses. His personal website can be found at http://people.brandonu.ca/nantaism/ and on he is on twitter @MikeN_bu

Reynold Redekopp PhD, taught junior high and high school for 28 years and is currently an instructor at the Faculty of Education, University of Manitoba. He began using personal computers in his second year of teaching and has been using them and studying the educational implications ever since.
Twitter and LinkedIn: @rredekopp
Blog: http://rredekopp.blogspot.ca/

Introduction

By Reynold Redekopp

Welcome to *Education and Technology: Manitoba Makers and Coders*. This is the second volume highlighting Education and Technology in Manitoba. The first volume was *Education and Technology: Manitoba Action and Reflection* in which authors contributed chapters about their thoughtful use of technology to enhance, and transform, teaching and learning. The focus for this volume is a bit tighter as we examine what is happening within the related phenomena of the Makerspace movement and the Coding in the classroom movement.

In this volume we are quite intentional when we use the term "Making' instead of Makerspaces as we think the concept has little to do with the space and everything to do with the mindset. Making is a way of thinking and this is the big idea for teaching and learning, not whether it happens in the library, classroom, gym, or art room. The other point that needs to be emphasized here is that Making is not about technology. There are lots of cool technologies out there that students can explore with, but the Maker idea is not dependent on them. Paper, glue scissors, cardboard, etc., can all be part of a very successful Maker way of thinking and acting.

The Coding[1] movement is curious in that few people can really agree on what this Coding part should entail and how to define it. The computer science people argue that coding is only a part of their program and that the emphasis should be on computational thinking, and could include some coding to illustrate the concepts. Many popular approaches to introducing Coding to younger students focus exclusively on the coding aspect with limited understanding of computational thinking. This makes Coding is a moving target at the moment.

And now these very current topics are addressed by fifteen Manitoba educators from K-12 and also in preservice classes. Making and Coding may be current buzzwords, but by highlighting the work these educators are doing we hope that the reader will see opportunity for taking the big ideas and using those to improve learning and teaching, whether the buzzwords fade or not. Both Making and Coding have value, but that value is dependent on the context of the teachers and students. These are not the 'silver bullets' that will fix education. They are great ways to think about how we teach and how students learn, and must be thoughtfully implemented only after we decide what is really important in the lives of our students.

I was able to teach a post-bacc course in Spring 2017 where we examined whether Makerspaces were a fad or the future. John Evans (Chapter 1), from Manitoba Education and Training, was a

[1] We use 'Coding' to describe the movement and 'coding' to describe the actual process of programming a computer.

guest speaker in our first class and when we asked him about the fad or future idea, he stated that if it is confined to libraries, it will be a fad. If it becomes a teaching mindset then it has a future. Fortunately we didn't stop the course at that point, even though we now had the answer! We visited five different schools to get their perspective on Makerspaces. It became clear very quickly that Making was the important concept, not the space. A few key components to Making also became clear: student voice, student choice, self-direction, passion, and the right to fail – and acknowledge failure as part of the learning cycle. In the Maker mindset teachers listen to student ideas for inspiration, give them choice in how they will approach problems, and avoid telling them the 'right way' to solve the problems. The teacher's task is to manage the context, and tease out the curricular objectives from the students' work. Shauna Cornwell addresses this more fully in Chapter 2 where she explains her school division's process of implementing a Maker movement and the thoughtful process that ensued.

In the first volume Mike Nantais, co-editor, noted: "Postman[2] (1995) wrote, 'A new technology does not add something, it changes everything.' We wanted authors to explore these ideas in their work." The chapters in this book are about challenging ourselves as educators, both to make changes that will help our students learn, but also to be thoughtful about what is gained and what is lost in the transition. This metacognition is useful personally, but also helps us understand why other educators might be hesitant to join us. We need to acknowledge the 'comfort zones' that we live in and how we can begin to take the risks required. Or, as Peggy Hobson so aptly puts it in Chapter 6, "Let's not just 'stay home' anymore." We also need to find our support groups. John Evans (Chapter 1) describes the online resource, Maple, that he has been instrumental in creating for teachers across the province to share and gather ideas in Making, Coding, and many other topics.

The Maker part of this volume is organized semi-sequentially from Early Years to Senior Years. The reader should be aware that most of the chapters are written by teachers who have been working on Making for a while now. Their programs have grown as they developed the ideas, gathered the materials, and listened to students. Anyone who wants to start should accept the fact that a program has to grow - often from very humble beginnings. Renee Sanguin (Chapter 3) describes the journey of their very successful Maker program. Success begins with asking the right questions. Amanda Ross outlines a grade one inquiry unit (Chapter 4) with some very practical ideas and insights. Chapter 5 is an incredible list of resources Amanda has developed for Early Years teachers looking for ideas and supplies. The list grew out of one of the assignments in the Makerspace course described above. Student choice is important and she chose to create this amazing resource.

---

[2] Postman, N. (1998). Five things we need to know about technological change. Retrieved from https://www.student.cs.uwaterloo.ca/~cs492/papers/neil-postman--five-things.html

Making can be a big attitudinal shift and it helps if administration is paying attention and making adjustments to help the process along. In Chapter 6 Peggy Hobson, a principal, describes her efforts to support the shift in her grade 5-9 school, and it all was kickstarted by some students with a passion for 3D printing. This got Peggy thinking and she set about restructuring the school day to foster more active teaching and learning. Another chapter in this age range is Chad Wilson's plan to create a Maker-philosophy in his grade 7 classroom (Chapter 7). Chad provides some very practical ideas as to how he wants his classroom to work, along with his philosophical orientation for Making.

At the high school level, David Gamble (Chapter 8) describes some of the shifts in thinking that have to occur for teachers to implement Making ideas across curricula. He conducted teacher interviews and researched the Maker movement and describes how various inquiry ideas come together. He also describes some practical projects that have been implemented in his school.

Leah Obach and Devon Caldwell (Chapter 9) merged Making and Coding into one chapter and so even though they work with young children, their chapter becomes the transition from Making to Coding. And, as they demonstrate, coding is not restricted to high schools. And Kirstan Osborne and Elizabeth LaPage continue the Coding emphasis with young students in Chapter 10 as they outline their K-12 approach to coding and computational thinking. The students move from a very physical, non-technology representation of problem solving to a high school program that makes failure an acceptable part of the learning process. We are reminded that learning to accept failure is not an easy process.

Finally we have a group of chapters that focus specifically on high school computer science. In the first, Chapter 11, Eleni Galatsanou Tellidis asks the pivotal question, "What is the difference between computer science and coding?" She then proceeds to examine different ideas about how and why we should teach computer science, especially for younger students. What do they really gain from this? These are critical questions we should be discussing before we implement anything. Chapter 12 is an approach that one of the editors, Reynold Redekopp, uses to encourage a different way of thinking about teaching computer science for his pre-service teachers. If it is true that we need more students to understand what is happening inside the machine, then we need to attract more students, particularly of underserved groups, to high school courses. One way to do this is to take advantage of modern technologies to make the courses more flexible and appealing. Matteo Di Muro describes how he takes this approach and combines it with Maker ideas to offer students more voice and more choice in his computer science classes (Chapter 13). Matteo is clear that this transition is sometimes difficult for students and illustrates how he scaffolds the process for them. Finally, Kate Nizio (Chapter 14) tells us about her concerted efforts to make computer science classes more exciting and accessible. She tries to remove the stereotype of the typical computer science student to draw in a more diverse group of students. Kate then makes a very deliberate effort to find ways to make computer science accessible to all students, especially when both the concepts and the coding becomes really challenging for the students.

Making and Coding are ways of thinking about problems and should be seen as such. They are not universal solutions. They are ways of approaching a problem, much like the scientific method. They should be seen and taught as such. There are many other ways of problem solving, some of which are much more holistic, and are effective in their own contexts. Our students need to learn to consider the strengths and weaknesses of various approaches to problem solving and use each approach as appropriate.

Mike and Reynold, co-editors, offer huge thanks to the authors of these chapters. Their contributions were well written, thoughtful pieces that required minimal editing on our part. You will see their passion come through in their writing. They managed to do their writing along with all their other responsibilities and we are deeply grateful. And we hope that you, the reader, will find inspiration in reading these chapters. If you have already tried some Making and Coding we hope you will find some new ideas here. If you are new to these Making and Coding, remember that none of theses authors started with a full-blown program. They started with an idea and some inspiration, found some others to join them, and built their program piece by piece. Hopefully you will find an idea to inspire you to begin.

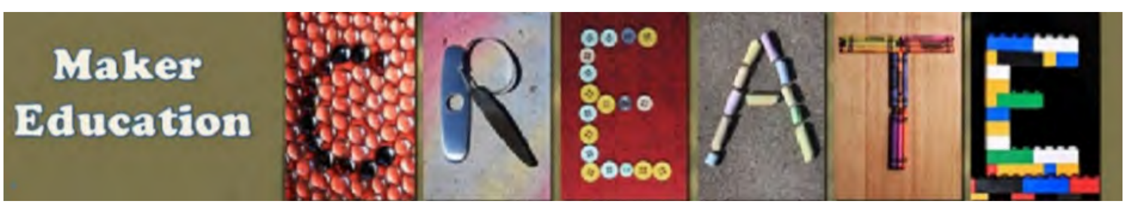
Chapter 1 - Maker Education and Maple: Connecting Educators in Manitoba

by John Evans
Manitoba Education and Training

Just a few short years ago if someone had spoken the words Making, Makerspaces, Maker Education or #makered I would have reacted with a blank stare. Each of these terms had a little or no meaning at all just a scant three years ago. Then in the spring of 2014 they seemed to be flying across all of my favourite social media networks. Fast forward to 2017 and these words are an integral part of not only my vocabulary but that of a very large portion of the teaching population across the globe.

I had been investigating this "Maker" phenomena on my own in early 2014, but it wasn't until I attended EdCamp Winnipeg in June that my "Maker" journey really started. It was here that I discovered that many of the Manitoba educators I knew via Twitter connections were investigating this Makerspace thing as well. It was also at this time that I first had the chance to chat with teacher/librarian **Brandi Bartok** and subsequently visited her library where I discovered her passion for makerspaces and making. That visit and conversation prompted even more research on my part until in very short order I was convinced of the value of this makerspace/making/MakerEd phenomenon.

The next step was to begin the development of the **"Maker Education"** group in **Maple**. This group would be dedicated to the many and varied forms of making and makerspaces that were gaining traction in schools across the globe and serve as a resource base for all Manitoba teachers interested in the topic. The group is designed to be a starting place for those interested in beginning with making in their schools. as well for folks who are well along in their journey to find helpful resources and ideas from other educators who are seeing the value of making in their libraries, classrooms and schools.



In the **Maker Education** group, as in other groups in Maple, there are two key sections of interest - the **Forum** and the **Wiki** components. Think of the Forum component as your news feed where you will find the latest items the group administrator has shared to the group. The

Wiki component can be thought of as an online binder where items are grouped according to various themes/topics. Some of the resources shared in the Forum may often be shared in the Wiki component as well if they related to one of the themes in the wiki itself.

Below is an example of a post to the Forum section. This fall I have been sharing a weekly blog post or two every Friday. I encourage you to investigate them each week.

If you join the **Maker Education** group or any other group in Maple, you will automatically receive an email update of what's new in the group each Friday.


MakerEd Discoveries to September 6, 2017

Happy First Days of School!!

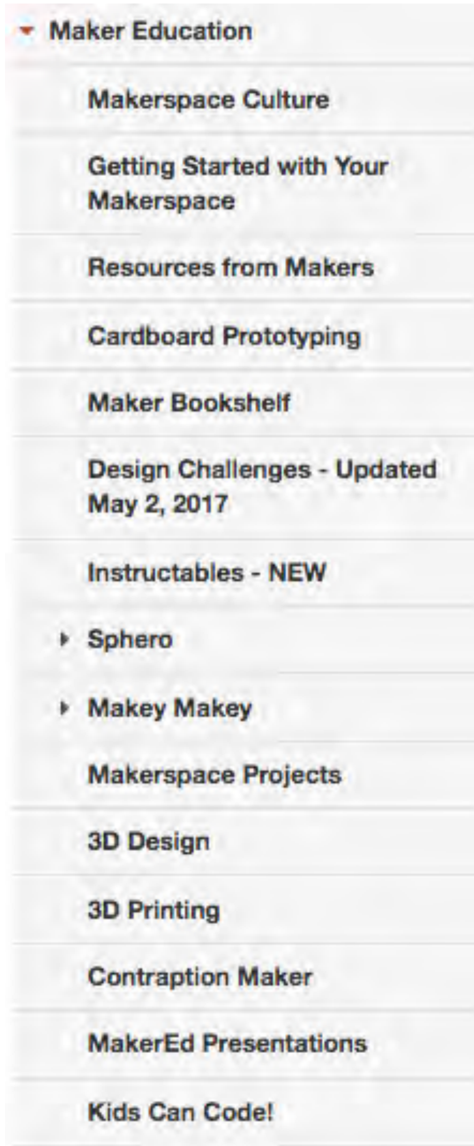
Here are a variety of Maker Education resources you may find useful in your classroom, library or makerspace!

12 Super STEM Learning Scenarios to Share with Your Students

Global Digital Citizen

****

The Wiki component of the Maker Education group has numerous sections covering a wide range of topics. These include Makerspace Culture, Maker Bookshelf, Design Challenges, 3D Printing and Cardboard Prototyping to name a few. The full page listing can be seen in the accompanying screenshot.



Each page contains images, embedded video, resource links and more to popular Makerspace educators and the concepts/themes taking place in makerspaces. The content of the wiki grows regularly so keep your eye on the weekly updates for new additions.

If you're just getting started and looking for information about the **Maker Education** movement and makerspaces or if you're seasoned Maker Educator, I encourage you to investigate the **Maker Education** group in **Maple**. If you have suggestions for posts or wiki pages please let me know as I'd love to investigate them further with you.

Maple is available to all Manitoba certified educators and school clinicians, pre-service teachers, Faculty of Education instructors and Manitoba Education and Training staff. Individuals can request a Maple account here: http://bit.ly/MapleSignUp

For more information about **Maple** please contact John Evans john.evans@gov.mb.ca or @joevans on Twitter.

John Evans is the Learning Technology Consultant with Manitoba Education and Training and the administrator of Manitoba's provincial online learning environment Maple - the **Ma**nitoba **P**rofessional **L**earning **E**nvironment, He regularly shares education innovations and best practices via the groups he's established in Maple. These include the Professional Learning, Kids Can Code, and the Maker Education groups.

Chapter 2 - A Culture of Creation

By Shauna Cornwell

Setting the Stage...Students as Makers

The "culture of making" is alive and well in Winnipeg School Division (WSD). Makerspace, or focusing on a "maker mindset" which involves learning and spaces where students gather to create, invent and learn is prominent in a number of WSD's 79 schools. These makerspaces are often found in libraries or other common areas. They are also set up regularly in classrooms and flexible in their design or use. The learning at WSD's two STEAM Centres follows parallel pedagogy and practice.

The learning can vary greatly and range from developing stories through Lego, to coding using an input/output device such as Makey Makey, to designing 3D imagery, to creating using stop motion animation, to inventing something brand new from take- a parts or cardboard. There are endless possibilities both high tech and low tech; utilizing imagination, creation, collaboration and innovation.

Makerspace environments allow for student-ownership, self-direction and problem solving by inspiring deeper thinking, with hands-on experiences, tinkering and constructing. With strong basis in the the theory of Constructivism (Vygotsky & Piaget), Constructionism (Papert) and Inquiry-based learning makerspaces offer students unique learning opportunities to construct their own understanding and knowledge of the world, through experiencing things, and reflecting on those experiences. Using a Makerspace model allows student to ask questions, find ways to answer those questions and drive their own learning. Within this context, teachers give learners the chance to develop a special skill set that is so necessary in today's world. These skills include critical and creative thinking, flexibility, innovation, risk taking and resilience. Makerspaces build essential skills and competencies which are embedded throughout the curricular areas.

Spaces where students make and create are nothing new to schools. Early years classrooms, libraries, art rooms, dance studios, theatres, band rooms, and industrial arts labs have served as "makerspaces" for our students at varying degrees and in a number of different ways for many, many years.

But what is new is the urgency to ensure that every classroom, encourages students to be creators and that moving forward ALL classrooms find an entry point in shifting from a passive learner model to one where students are at the centre in a stance of active learning. This is essential as we strive to meet the diverse needs of our learners in our ever changing world. We know we need to change our focus and our approach in order to meet our learners needs and the demands of our society. We need to support our students in becoming the globally aware, creative, adaptive, resilient, digitally fluent, flexible thinkers necessary in today's reality. Initiating programming that prioritizes students as makers is one opportunity to do so.

Research shows that as children move up through the grades engagement levels decrease significantly. The 2015 *Gallup Student Poll, Engaged Today, Ready for Tomorrow* defines engagement as student "involvement in and enthusiasm for school." After surveying over 900 000 US students the study shows engagement levels steadily drop as students move from Grades 5-12. The poll indicates that while over 75% of 5th graders are engaged in school, this numbers steadily drops to 34% of students being engaged in their learning by the 12th grade.

The Canadian Education Association (CEA) 2011 study looking at intellectual, social and institutional engagement in schools entitled, "What did you do at school today?" shows similar trends. These decreasing levels of engagement tell us something has to shift. Engaging our students through active learning and providing learning opportunities that encourage both voice and choice is an avenue of change.

In response to this, in addition to shifting needs in the way of employability skills, we see more and more schools considering more student-centred, student driven learning initiatives, such as a focus on "making" and student creation to support more student engagement and investment in learning.

Maker ~~Space~~ Culture

This may look like schools establishing a more formalized "makerspace" in libraries, learning commons or other shared spaces for logistical reasons. Materials in a makerspace can vary greatly from low tech and often inexpensive (building supplies, cardboard, take-a-parts) to high tech and more expensive (3D Printer, tablets, circuitry, video equipment etc.) Schools must find a model of making that works best for their context and their students within their school community both financially and logistically.

This maker culture focuses on design and creation that empowers students and is not dependent on location. In the words of Brian Aspinall, educator and maker:

> "Maker is more about culture and less about 'space' and 'stuff'[3]".

Ultimately, all classrooms should be a flexible "makerspace" over a given day or perhaps week. Learning by **doing** and **creating** is our best way to "teach kids HOW to think, not WHAT to think" (Mead, 1942).

---

[3] http://brianaspinall.com/maker-is-a-culture-not-a-space/

Active Learning

Establishing a making culture is about deep, contextual learning using the creative process, and the thinking or understanding that is gained through this active learning. When we rely too heavily on a "sage on the stage" teacher-centric model of instruction and consumption, it is too often the *teacher's voice* that plays an overly dominant role. When we move to more of a "guide on the side" learner-centric model of creation we activate our students in the learning process and allow *their voices* to be heard.

In reality, both models have a place in the learning lives of our students. There is a time and place for a more lecture style or story centred approach and a time and place more suited to a hands on model. The trick is to find the balance and opportunities best suited to your context.

Here are some entry points or examples of shifting toward *Student as Active Learner or Maker*:

Student as Passive Learner	Student as Active Learner or "Maker"
Students read assigned chapters of a novel, and do end of chapter questions or Teacher reads a picture book, class discusses the protagonist's problem and predicts solutions that might occur later in the story	Novel or Picture Book Engineering[4] • STEP 1- Student or class reads the book • STEP 2- Student creates a character sketch or profile • STEP 3- Student identifies a problem or goal the character faces • STEP 4- Student ideates 3-5 solutions for the character's problem • STEP 5 - Student chooses one and creates a prototype
Teacher provides information information on an Ancient Society in a lecture format. Students read sections in a textbook and answer questions in their notebooks.	Students are given some guiding questions to investigate on an Ancient Society. In groups they determine an area of interest and decide on what they are going to create to show understanding of their topic (3D model, video, art pieces, skit etc) They present their learning to the class.

[4] http://www.novelengineering.org/books

Teacher does a 40 minute lesson on the board on a math topic. Students are assigned 20 textbook questions as homework.	Students are given a math topic to investigate after the teacher does an introduction. They are tasked with watching videos from a list of resources (i.e. Khan Academy) to build understanding then given a handful of problems to solve. They are asked to show their thinking and create their own video using an app like Explain Everything or other screencasting tool.
Teachers shows videos and demonstrates different types of simple machines. Class discusses examples. Students do worksheets on simple machines.At the end of the unit there is a test.	Students ask questions about and then investigate simple machines. As a group, they are tasked with designing a Rube Goldberg machine to complete a simple task, that includes 3 simple machines, which they must explain in their video documenting their learning.
Research Project- students are assigned a topic to investigate, given a list of questions to answer and told what format they must present in.	Genius Hour- students are invited to take on a project of their own choosing based on their own individual questions, interests, and/or strengths. With guidance they plan all elements of their own learning focus.
Teacher scribes the story of a reluctant writer during Writer's Workshop after the teacher has done a mini lesson on parts of a story .	Student uses Lego Story Starter to build a beginning, middle and end of story. Student takes pictures of lego scenes using an iPad and inputs the pictures into an app such as Adobe Spark. Student tells his or her story orally to go with pictures and creates a video in minutes.

Shift in Thinking and Planning

Putting students in the active learning stance of "maker" requires a shift in thinking and planning for teachers. It means setting the stage for learning, without being a one-person show, but instead letting students in on the act. Often teachers plan a lesson, or unit with the intent of meeting certain, specific learning outcomes within that curriculum. When we give students a more active role in their learning, through creation, we instead plan for offering rich learning opportunities and then have to skilfully find all of the outcomes students have met, often cross-curricularly. It sometimes means investing more time in a topic you may have typically gotten through a lot faster with more traditional means and teacher direction. It sometimes means not focusing so much on how to "cover" the curriculum but instead really taking an interdisciplinary lens to seeing the deep learning that is happening cross-curricularly. It sometimes means letting go of some control and instead imagining the possibilities.

Shifting Role of Teacher

When we give students more opportunities to learn through creation, these learning experiences ultimately shift the role of teachers as they step back into more of a role of facilitator or coach helping to guide students in the making process. However, teachers still play an essential and active role in supporting students in their learning journey by:

- setting criteria (with students)
- supporting skills
- guiding practice
- encouraging and modeling reflective practice
- scaffolding students through the design process

4 Key Pillars of Makerspace

This section will explore the **4 key pillars** that serve as foundational, pedagogical pieces to set the stage for establishing a "maker culture" in your classroom as a powerful and effective learning environment:

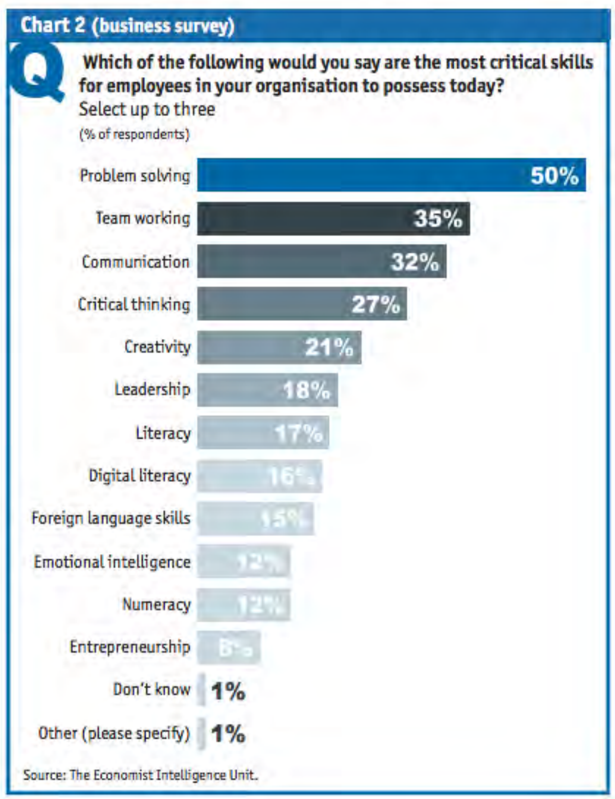
1. **4 C, Competency Focused, Deep Learning**
2. **Design Thinking Process**
3. **Personalized, Active, Inquiry Based learning**
4. **Reflection & Metacognition**

Pillar 1 - 4C, Competency focused, Deep Learning

Critical & Creative Thinking, Communication, and Collaboration.

When we create rich learning experiences for students that encourage them to think, communicate and work together, we offer them the opportunity to learn in a way that is both engaging and empowering. With shifting times in society come shifting needs as learners. We can look to research about employability skills moving forward to drive the decisions we make for our students. The graphic below, based on a study done by the Economist Intelligence Unit is a clear indication of how learning focused on developing the four competencies, helps prepare students with employability skills they need for the future. Although we recognize we still need to prepare kids as literate, numerate members of society, there are also a number of other skills we must help support and develop in students. If schools' ultimate goal is to **prepare students**

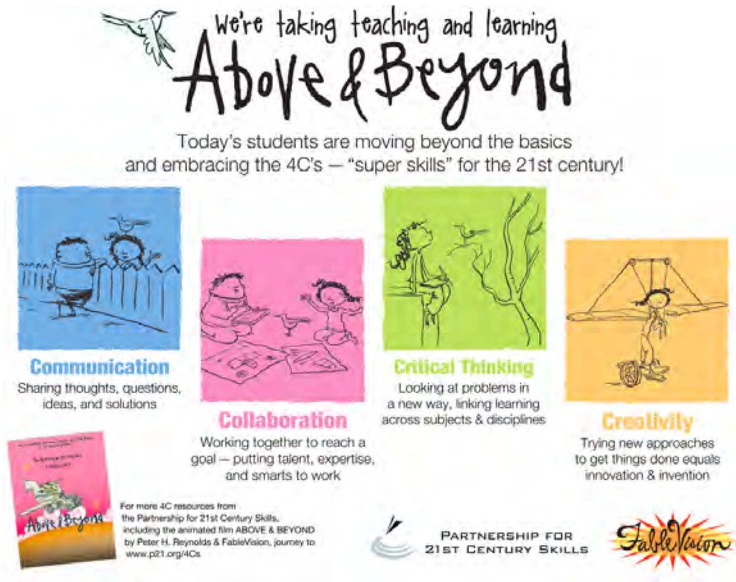
for the future then we know programming that focuses on problem-solving, teamwork, communication, critical thinking and creativity through *making* will help get them there.



[5]

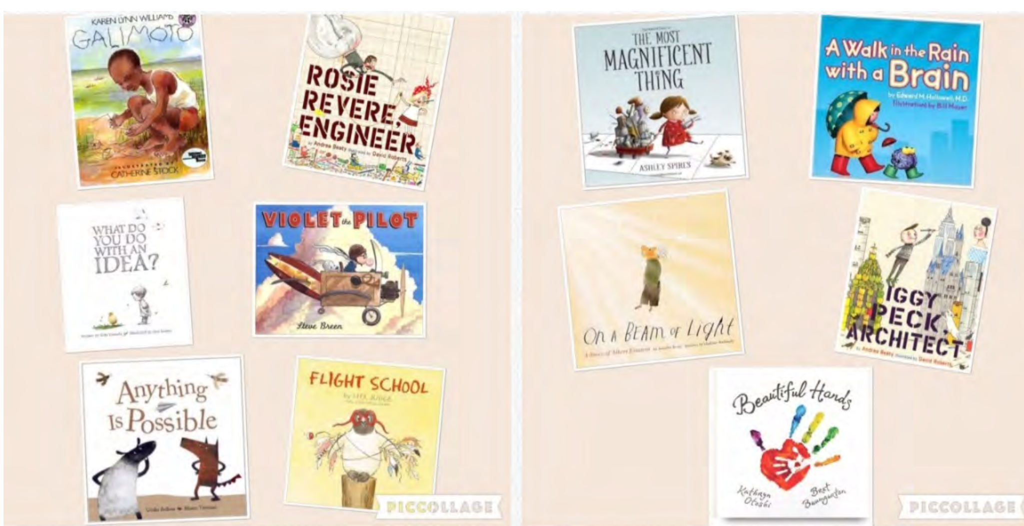
One resource we can look at for guidance is the Partnerships for 21 Century Learning (P21) organization and the 4Cs poster featured below[6]:

[5] https://www.eiuperspectives.economist.com/sites/default/files/Drivingtheskillsagenda.pdf
[6] http://www.p21.org/storage/documents/4csposter.pdf



[7]

An important place to start is to make sure we, as educators within our schools, and, our students, have a common understanding of what these four words mean, along with what they look like and sound like in practice. The video entitled Above and Beyond also by P21, is a good starting point for beginning conversations with students. Many classroom teachers also start by using picture books to prompt conversations around the 4Cs.



Ultimately, we want students to gain insight into what it means to be a skilled *critical* or *creative thinker, collaborator and communicator.* We need to give them the language they need to build this experience base and understanding. We can look to the British Columbia Redesigned Curriculum, which began implementation in 2016, as a model for learning with a strong

[7] http://www.p21.org/storage/documents/4csposter.pdf

foundation of a competency focus. The competencies that serve as a common thread from kindergarten to grade 12 are:

1. Thinking

• Creative Thinking • Critical Thinking

2. Communication

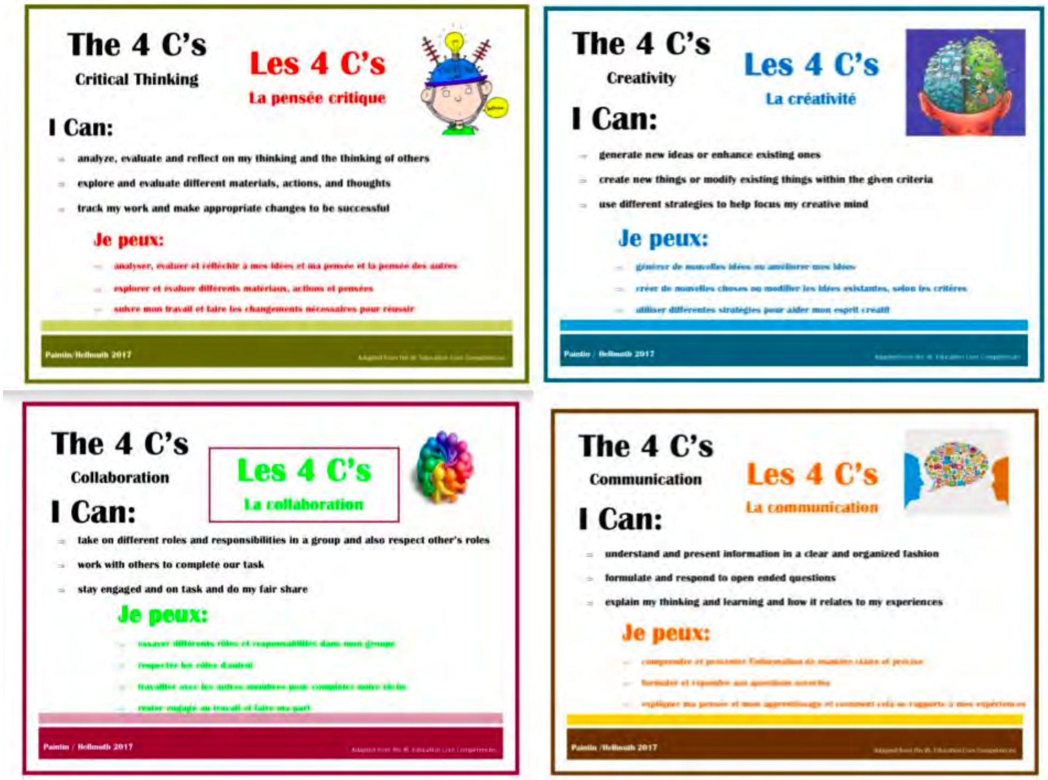
3. Personal and Social

• Positive Personal/Cultural Identity • Personal Awareness and Responsibility • Social Awareness and Responsibility

The Province of BC's rationale for the focus on competencies is explained here:

> *At the heart of the of BC's Redesigned Curriculum are the Core Competencies, essential learnings and literacy and numeracy foundations. All three features contribute to deep learning. Core competencies underpin curricular competencies in all areas of learning. They are directly linked to the educated citizen and as such what we value for all students in the system.*
>
> *Competencies come into play when students are engaged in "doing" in any area of learning. This includes activities where students use thinking, collaboration, and communication to solve problems, address issues, or make decisions. The ultimate goal is for learners to employ the core competencies every day in school and in life, and for the core competencies to be an integral part of the learning in all curriculum areas.*[8]

[8] https://curriculum.gov.bc.ca/competencies

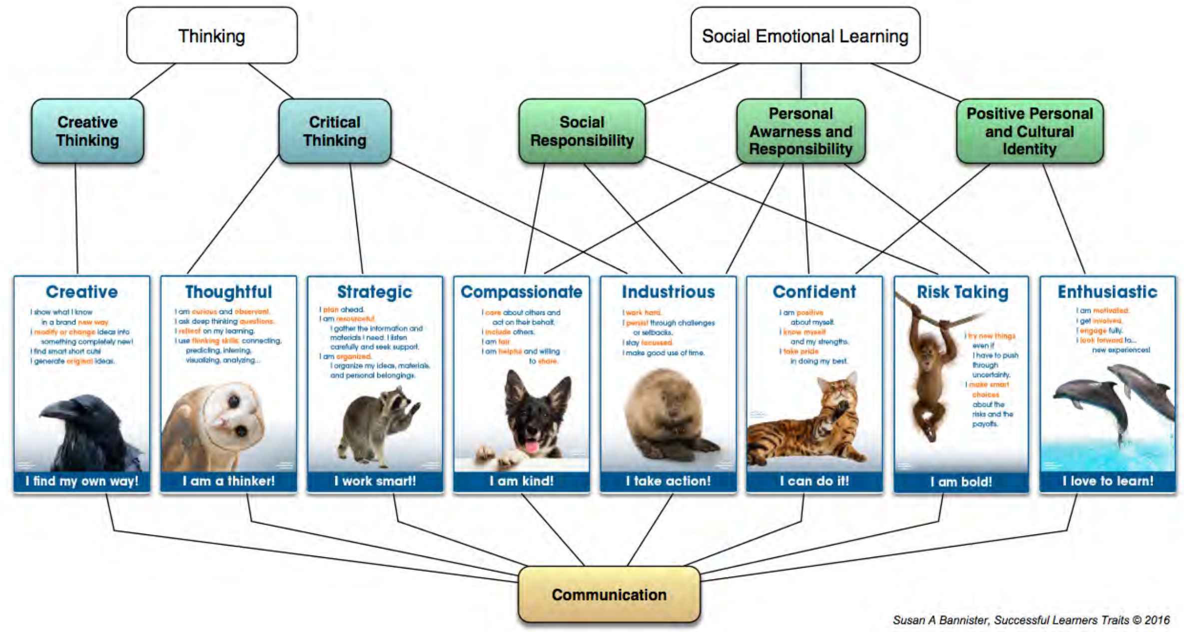


[9]

The posters above are based on the [Learner Profiles](#) established within the BC Competencies and are used to help students self assess and use "I Can" language in WSD's STEAM Centres, which will be explored later in the chapter.

Another excellent resource to support student understanding around learner skills and competencies is the [Successful Learner Trait Framework](#) created by Sue Bannister, an educator in Comox, BC.

[9] http://blogs.winnipegsd.ca/steam/pillars-of-steam/the-4-cs/



[10]

Sue Bannister has developed posters, assessment resources and other classroom tools to support teachers and learners in supporting student learning around the competencies.

After developing students understanding into what the competencies look like and sound like in practice, we need to next provide learning opportunities that allow them to practice these skills. A focus on communication, problem solving, creativity, critical thinking and other competencies is not new to Manitoba curriculum and reporting. We can look to a sample of subject areas reflected in this Manitoba Grade 1 to 8 report card[11] below to see how the learning focus within a making environment supports learning across the curriculum in interdisciplinary ways.

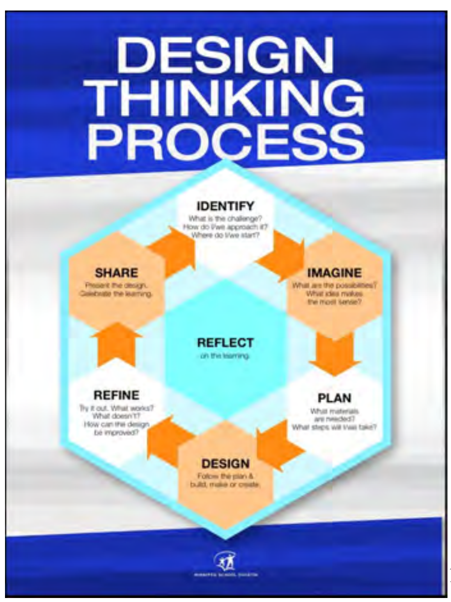
[10] https://www.successfullearners.ca/
[11] http://www.edu.gov.mb.ca/k12/assess/report_cards/index.html

Social Studies	Teacher:	
☐ EAL ☐ IEP		Term 1
	Knowledge and understanding	
	Research and communication	
	Critical thinking and citizenship	
Science	**Teacher:**	
☐ EAL ☐ IEP		Term 1
	Knowledge and understanding	
	Scientific inquiry process	
	Design process and problem solving	
Mathematics	**Teacher:**	
☐ EAL ☐ IEP		Term 1
	Knowledge and understanding	
	Mental math and estimation	
	Problem solving	
English Language Arts	**Teacher:**	
☐ EAL ☐ IEP		Term 1
Comprehension	Reading	
	Listening and viewing	
Communication	Writing	
	Speaking and representing	
	Critical thinking	

Pillar 2 - Design Thinking Process

Identify → Imagine → Plan → Design → Refine → Share

←Reflect→



[12]

[12] http://blogs.winnipegsd.ca/steam/pillars-of-steam/the-design-thinking-process/

Driving competency based learning is students thinking and "doing". Design thinking is a creative process to support our students' thinking and doing. It can be used to approach a problem/task that serves as a model or roadmap for students. It has universal applications across all subject areas and disciplines, and applies to all that students are creating whether it be;

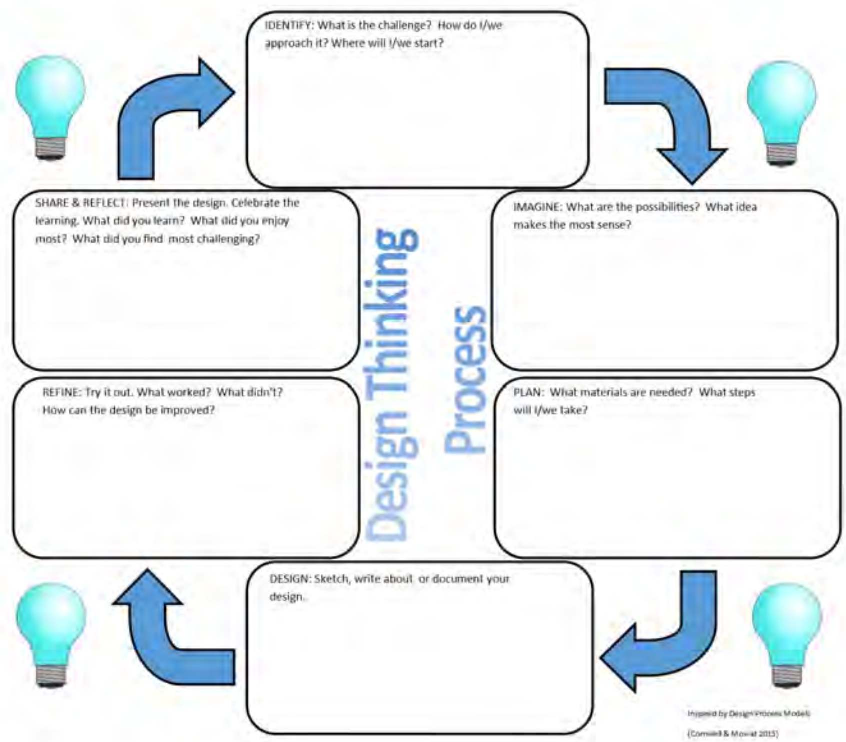
- creating a sculpture
- solving an ongoing classroom issue
- planning a campaign for a school wide initiative
- building a tower
- writing a poem or essay
- coding a simulation
- devising an experiment
- designing something using 3D imagery

Here is a look at each step of the process, a little more in depth:

IDENTIFY Students identify the challenge whether it be solving a problem or completing a task, which may be determined by them or provided by the teacher. Students think about where to start, what they already know and what they are wondering.
IMAGINE Students consider all of the possibilities. They generate ideas. They ask questions and find answers using a variety of sources. They ask more questions and find more answers. They generate more ideas and determine what idea makes the most sense.
PLAN Students plan for resources or materials they may need and what steps they will take.
DESIGN Students make, create or build following through on their design plans.
REFINE Students try out and/or review their design plans. They consider what works, what doesn't and make changes and improvements as needed.
SHARE Students share their work both locally and globally. They get further feedback and celebrate their learning.

REFLECT
Students think critically and constantly evaluate what's working and what's not. They make revisions to their design and refine their plan as needed. They gather feedback. They demonstrate resilience and tenacity as they face obstacles and setbacks. They problem solve and plan for next steps.

When first using this process it may be necessary for teachers to model and scaffold each step of the process as needed, just as one would with any new approach. The graphic organizer below can help students in the planning stages of a design project and help keep their learning visible.



[13]

Early years teachers appreciate the simplified model of a design thinking process below, which comes from the book, *Invent to Learn* from Sylvia Libow Martinez & Gary Stager to use with our littlest of learners:

[13] http://blogs.winnipegsd.ca/steam/pillars-of-steam/the-design-thinking-process/

[14]

rmission)In today's dynamic world we not only need students who are problem solvers, but who are also problem finders. Stanford's d.school whose goal is to "help you to use design to make change where you are[15]" uses the five steps of the design process below that begins with approaching all problems through the lens of empathy. This is an important skill as students move into the world of work and must learn to empathize with customers, clients or co-workers in confronting and addressing real world problems. Teaching our students to empathize with others through the guise of problem solving and design is an important skill.

EMPATHIZE → DEFINE→ IDEATE→ PROTOTYPE→ TEST

Using design thinking as a process may be used by students to drive a task assigned by a teacher, or to tackle a challenge the student has identified as a need. Regardless of whether students are using design thinking to build a structure, construct a piece of art, design a PSA (public service announcement) to address a community need, create code to program a device, or develop a solution to a math problem, the steps of the process remain the same as does the powerful opportunity for learning.

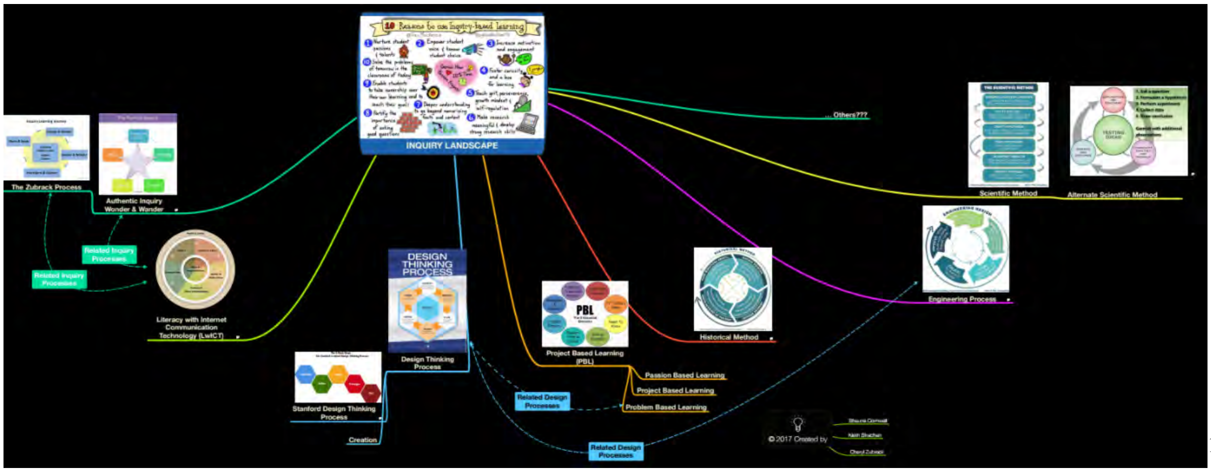
---

14 https://inventtolearn.com/

15 https://dschool.stanford.edu/

Pillar 3 - Personalized, Active, Inquiry-Based Learning

Design thinking is a form of inquiry based learning. It is yet another entry point in a student-centred, inquiry driven approach. We can look at inquiry as an umbrella term that encompasses many different entry points for student driven learning; others being, *the scientific method, project based learning, engineering process* or a more *wander and wonder type of question driven inquiry*. All models support a more learner focused, active stance of learning which often, at some point, has students creating or producing something to share their understanding.



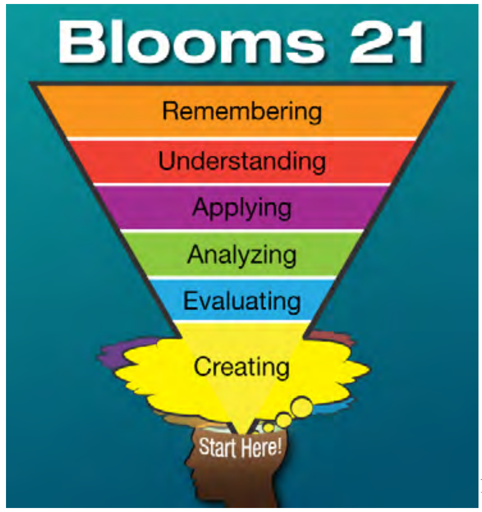
[16]

But with design, the creation process begins up front, serving as the driver for learning. Rich questions and deep learning then come from an authentic place of needing to know. Design thinking is not only about the final product but also about the PROCESS.

In 2012, Shelly Wright, an educator from Saskatchewan, wrote a blog post entitled, *Flipping Bloom's Taxonomy* in which she explores a 21st century version of Bloom's Taxonomy that puts creation at the forefront.

---

[16] https://themeaningofmerakicom.files.wordpress.com/2017/11/inquiry-mindset-1.pdf



[17]

This is an interesting perspective and although some checking of prior knowledge and upfront learning may need to happen before design and creation can take place, it serves to remind us of the necessity of pushing kids to think at higher levels and the importance of carving out the time to get to higher order thinking which can occur through creation.

Using the design thinking process as a form of inquiry based learning is also an exciting opportunity to allow student **strengths, skills, interests,** and **passions** to drive learning experiences. This shift towards a strength based model, as opposed to a more deficit focused model which can often drive some of what we do in schools is an opportunity to empower students and support them in becoming more autonomous learners.

Pillar 4 - Reflection & Metacognition

An essential component of establishing a makerspace and a culture of creation is that students have the opportunity to think deeply about their own thinking, doing and learning. Metacognition is defined in simplest terms as "thinking about your own thinking." The phrase was termed by American developmental psychologist John H. Flavell in 1979, and then further developed throughout the 1980s. [18] *The engine of metacognition is reflection***.** Thinking about our own thinking and the impact that has on moving learning forward is imperative for us as educators and it is imperative for our students.

As educators we must be reflective practitioners ourselves and model these practices FOR and WITH our students. We need to consider and plan for the most effective ways to support our students in reflecting on their own learning in meaningful ways.

[17] https://shelleywright.wordpress.com/2012/05/29/flipping-blooms-taxonomy/
[18] https://www.knewton.com/resources/blog/ed-tech-101/ed-tech-101-metacognition/

Students need the opportunity and the time to think about *where they have come from, where they are, and where they are going*. We want them to not only reflect on what they have done but to develop more awareness of "self as learner" by thinking about how they got there and the why behind the choices they have made. We need to support students in getting past reflection sounding like a play by play of the steps they took in completing their work or task.

Often when we ask students to reflect it is in the form of writing, which at times may paint a better picture of where our students are at as a *writer,* as opposed to his or her skills or deeper thinking as a *reflector*. **Personalizing reflection** options, based on a student's strengths and preferences allows us to truly get a glimpse into the reflective thinking our students are capable of.

In schools today, there are many more tools and options to support more **personalized reflection** and the documentation of learning. Reluctant writers may find a more attractive and effective means to share their thinking. Students can find the best fit for themselves when other opportunities for reflection are offered through blogging, video, images, audio files, graphics, and a variety of digital portfolio options or creation apps.

Here are some **personalized** options students can use to reflect on their making experiences:

Seesaw as a digital portfolio tool - Students in Grades N to 6 in WSD have access to Seesaw as a tool to support reflective practice. Student can document their learning and their journey through the Design Process using the 4C's with the built-in tools that allow for the insertion of *images, videos, drawings, voice overs* and *written text* all within the tool itself. Seesaw also provides the opportunity for an authentic audience for student work with both connection and communication with families made easy. For teachers, links to learning outcomes and formative assessment strategies, such as peer feedback is also built in.

Sketchnoting - we have seen sketchnotes become more and more prolific in the world of education in recent times. Sketchnotes are *rich visual notes created from a mix of handwriting, drawings, hand-drawn typography, shapes, and visual elements like arrows, boxes, and lines*[19]. Some students truly appreciate the opportunity to reflect using this very visual means.

Powerful Apps and Digital Tools - there are a number of great apps or web based platforms that can also support the documentation of the creative process and reflection in a personalized way.

- **Adobe Spark** [20] allows students to create and share visual stories of student learning over time including simply made, animated videos students of all ages can voice.

[19] Mike Rohde, The Sketchnote Handbook
[20] https://spark.adobe.com/

- **Explain Everything** allows students to explain concepts, and reflect on ideas using an interactive whiteboard and screencasting tool with real-time collaboration that lets students animate, record, and annotate.[21]
- **Book Creator** or similar e-book creators allow student to document their learning process through photos and simple accompanying text.
- **Video-** capturing video of students work in progress or students reflecting in their work in a sort of video diary, is a powerful way to capture student reflection.

Offering options and a more personalized approach to reflection is only one piece of the puzzle. First and foremost we must ensure we give ourselves permission to carve out the time necessary to allow students to reflect in meaningful ways. In the early stages this may take a lot of modeling and providing scaffolds for students, such as prompts and sentence starters using reflective language. Students need frequent reminders to talk less about the WHAT of their learning and instead focus on the HOW, WHY and NEXT STEPS of their learning.

One of the most powerful tools schools have used in teaching students the true art and science of reflection is Instant Challenges, a term that comes from Destination Imagination, an organization aimed at teaching students the creative process. An instant challenge is a challenge that teams of students are asked to solve in a short period of time without knowing ahead of time what the challenge will be. The challenge puts the team's collaborative problem solving abilities, creativity and teamwork to the test.

In establishing a classroom culture of creation, focused on competency based learning and design thinking, instant challenges are a great place to start with students! They provide an excellent entry point into reflective practice and documentation of student thinking, doing and learning.

Providing students the opportunity to learn through creation is essential in today's context. A focus on competency based learning, design thinking, reflection and establishing a student driven, inquiry-based learning environment are essential ingredients in ensuring that "culture of creation" is a success.

Building a Culture of Creation

We can look to the model of Winnipeg School Division's two STEAM Centres for establishing a making culture that drives all programming.

Housed in Pinkham and Rockwood Schools in Winnipeg since 2014, the WSD STEAM Centres support staff and students in WSD's 59 elementary schools. Over the course of the year, nearly 20 schools are serviced over each of the three blocks which run, October to December, January

[21] https://explaineverything.com/education/

to March and April to June. During those blocks, three students from each school visit STEAM one day a week participating in STEAM centred experiential learning activities focused on;

- An interdisciplinary approach using through the lens of S.T.E.A.M….Science, Technology, Engineering, Arts and Math
- Natural integration of technology
- Creation based, student focused, active learning
- Gaming
- Coding and Programming
- Challenge learning such as Instant Challenges and Rube Goldberg
- Regular reflection time build in throughout the day

Over the eight weeks the STEAM teachers are in regular contact with the home schools sharing learning and exposing classroom teachers to the practices, pedagogy and thinking that drive learning in the STEAM Centres. Connected learning is a key component of the STEAM program as the teachers and students build networking skills using platforms such as Twitter, Instagram, Edublogs, Spotify, YouTube, Skype and Seesaw. STEAM Teacher Jon Paintin, has worked hard over the last number of years to ensure home schools, classrooms and parents are all kept in the loop about the powerful learning happening in the STEAM Centres. Teachers also attend a professional learning session about teaching and learning using a "making" model and, after the eight week block, the STEAM Centre teachers then follow up in the home schools bringing the STEAM learning experiences to whole classrooms, back at the home schools. In 2016 the STEAM Centre programming was expanded to include French language programming one day a week for WSD's French Immersion and Milieux students, facilitated by STEAM teacher Tina Hellmuth.

STEAM or maker type programming continues to build in schools across Winnipeg School Division and on the larger educational landscape as a whole. To find out more about the maker movement in WSD be sure to read the next chapter entitled, "Our Journey to Makerspace" written by Renee Sanguin, celebrating the learning journey and accomplishments of the Ecole Victoria Albert Learning Commons Makerspace.

Putting students in the place of "maker" is a powerful opportunity for our 21st C learners to be creators, critical thinkers, problem solvers, collaborators, communicators and most importantly life-long learners. Building a culture of creation reminds us, that in the dynamic, changing world that is our reality, what we create drives what we do and where we go next. Learning is a journey, not a destination.

References

3. Image used with permission from The Economist Intelligence Unit
4. Image used with permission from Partnership for 21st Century Learning: Above and Beyond

7. Image copyright of Winnipeg School Division - used with permission
8. Image used with permission Susan Bannister.
10. Manitoba Education and Training: Assessment & Evaluation: Grade 1-6 Report Card
11. Image copyright of Winnipeg School Division - used with permission.
12. Image copyright of Winnipeg School Division - used with permission
13. Image used with permission from Gary Stager & Sylvia Libow Martinez
15. Image created by Keith Strachan, Cheryl Zubrack and Shauna Cornwell copyright Winnipeg School Division - used with permission
16. Image used with permission from Shelley Wright.

Shauna Cornwell- Mother. Learner. Maker. Friend

Shauna serves as the Enrichment & Innovation Consultant in Winnipeg School Division. She supports innovative teaching and learning practices, programming and initiatives across WSD. Prior to that she was a support teacher for ten years in the areas of inquiry, technology, library and learning, as well as a middle years classroom teacher for seven years. She is also the Affiliate Director for Destination Imagination in Manitoba.

Email: scornwell@wsd1.org   Twitter: @slcornwell   Blog: The Meaning of Meraki

Chapter 3 - Our Makerspace Journey:
Transforming the Library Learning Commons

by Renee Sanguin
Inquiry Support Teacher
École Victoria-Albert School

It all started with an invitation from our Library Consultant, Kevin Mowat: "Might you be interested in piloting a Makerspace program in your library?"
"What is a Makerspace?" we wondered.
So the journey began…

As a group of teachers, we were, and continue to be, interested in student voice, student learning, inquiry and reflective practice. When presented with the opportunity to explore Makerspace we had many questions. How might Makerspace make our students better learners? How do we help our students take their learning deeper during Makerspace? How will newcomer EAL students integrate? How might students apply and connect design process and the 4Cs to the real world? Will my students just play? How do we explicitly teach the 4Cs? How will I know that my students are learning? How do we create meaningful experiences for all of our learners? How do we keep learners at the heart of all we do in a world where knowledge and information is so easily accessible? How do we create equity, accessibility, and personalized learning for all our students?

We looked at our library and wondered if we could shift the school culture to think about this space as a Library Learning Commons, a Makerspace, a place for the Design Thinking Process and for practicing the Deep Learning Skills of collaboration, communication, critical thinking and creativity.

Our School Community

École Victoria-Albert School is located in West Alexander, an inner city neighborhood in Winnipeg. This is one of Winnipeg's original mixed-use residential, commercial and industrial areas with many buildings and services dating back to the late 19th and early 20th centuries. Over 4,000 people call West Alexander home; it is made up of transitional and low rental housing and maintains one of the highest immigrant populations in Winnipeg. Many newcomer families settle in the area's transitional housing such as IRCOM (Immigrant and Refugee Community Organizations of Manitoba), where the maximum tenancy is three years. This creates a constant level of mobility in our school community as newcomer families arrive, transition, and eventually relocate to other areas of Winnipeg.

Facts about inner city families: (2011 Census)

- approximately 57.1% of the households are single parent families, as compared to the City of Winnipeg average of 46.3%;
- 31.8 % have less than a Grade 9 education compared to the Winnipeg average of 19.8%;
- 46.1% are immigrants as compared to 21.7% in Winnipeg;
- unemployment rate, according to 2011 Census, is 9.9% compared to a Winnipeg average of 5.9%.

Our school community is comprised of about 400 students, from Nursery to Grade 6. We are a dual track school with French Immersion and English classrooms, as well as a Learning Assistance Center, (small classroom setting for behaviour support). About 70% of our students are EAL, 25% are Indigenous and we have enrolled 60 Syrian refugees since January 2016.

Innovation in Program Design

In the initial stages of our Makerspace inquiry we needed to learn everything we could about Makerspace. We read books such as, *Invent to Learn* (Martinez & Stager, 2013) *Worlds of Making* (Fleming, 2015) and many digital resources such as *Makerspace for Education* which helped us to understand the pedagogy of Makerspace through the work of Seymour Papert. We sought out our Innovation and Enrichment consultant Shauna Cornwell and our Library Consultant at the time, Kevin Mowat, to help us deepen our understanding through professional development. We met often as a group.

It was January 2016 and we felt we now knew enough to start. Initially our program focused on using *Instant Challenges* as a way to teach our students about collaboration, critical thinking, creativity and communication. As a response to our diverse cultural community, communication and collaboration were critical in order to help our students integrate and feel like part of our learning community.

We decided to use the library as a place to meet with students, thereby shifting our collective thinking from a traditional library setting to a Library Learning Commons that honours the interconnectedness of all work that happens in school. We made innovative choices around furniture, such as flexible seating and differing table heights, and we introduced new ways of interacting with each other and with the space. We organized shelves with LEGO and other building materials such as K'Nex, DUPLO, BRIO, TINKER TOYS, Perler beads and sewing machines, knitting needles, recycled materials and a set of Rig-a-ma-Jig to inspire, engage and empower our learners.

These simple changes offered our students time to collaborate and communicate and do work that promoted critical thinking and creativity through interaction with physical materials and technology. Through these changes, we felt that we had started to make the shift towards the innovative learning environment necessary to enhance our journey to learner empowerment.

Pedagogical Documentation and Reflection

One of our professional learning goals for the year was to explore student voice through pedagogical documentation. We wanted to figure out what Makerspace might tell us about student learning. We knew we needed to explore and debrief with students and teachers engaged in our Makerspace journey. This professional inquiry centered around our curiosity about Makerspace, Inquiry, Student Voice and Pedagogical Documentation.

The "Vic-Al Makers" use the design thinking processes to frame much of what we do. Debriefing after all our Maker work is a substantial component of each learning opportunity in our Makerspace. In collaboration with students and teachers we often used a modified version of the Visual Thinking Strategy called Makerspace Metacognition, the 4 C's (collaboration, communication, critical thinking, and creativity), regular debriefs, and co-created criteria/checklists to thoughtfully plan for this learning.

To help us deepen our understanding about pedagogical documentation we turned to the experts at Harvard University involved in Project Zero. We explored such professional resources as Making Thinking Visible by (Ritchart, Church, & Morrison, 2011) and Visible Learners (Krechevsky, Mardell, Rivard & Wilson, 2013) to help frame our work with reflective practice. Other professional resources, such as Natural Curiosity by the ICS Lab School in Toronto and Connecting the Dots (Kozak & Elliott) at Learning for a Sustainable Future, were foundational material for us in our Inquiry studies from previous years.

In documenting the Makerspace movement, we began by asking (as outlined in the book *Visible Learners*):

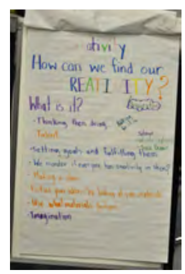
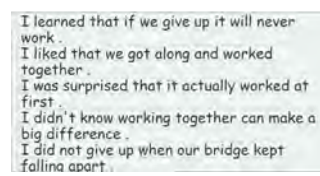
What is going on here?

What feels surprising about this moment?

What does this tell me about what these learners know and care about?

Our hope was that the documentation would provide new vantage points on learning, allowing all learners (students and teachers) to see knowledge in new ways and, in this new way of seeing, explore the possibility of being able to affect some sort of change on it. We hoped that documenting the learning would allow all of us (students and teachers) to communicate the experiences captured and the knowledge gained in a way that might build collective knowledge around Makerspace and what happens during Makerspace times. We wondered if this kind of documentation might help all participants (students and teachers) connect to each other as learners and human beings. We also wondered about the role of making learning visible…creating spaces and places for documentation as an ongoing process that could be revisited and added to as we moved together to develop our understanding. We were making the observations intentional and stepping back with a sense of curiosity.

As a result, we spent a number of months gathering documentation through interviews, digital pictures, whole group debriefs and reflections recorded on chart paper and anecdotal conversations with anyone who was interested in talking about their learning. In the Instant Challenge and Makerspace context, we became what is called in the book, *Visible Learners*, the "Memory of the Group". Serving as memory keeper allowed us to revisit, celebrate and share our learning with others.

Whole Group Debriefs	
Student posts on Seesaw	I learned that if we give up it will never work. I liked that we got along and worked together. I was surprised that it actually worked at first. I didn't know working together can make a big difference. I did not give up when our bridge kept falling apart.

VTS strategy in the Makerspace context What do we think is going on in this picture? What do we see that makes us say that? What more can we find?	

Makerspace Metacognition in Depth

Metacognition is essentially *thinking about thinking.* We want children to *think about their thinking,* have a common language to *talk about their thinking,* and *discuss strategies they use* to understand.

In Visual Thinking Strategy (VTS) discussions, teachers support student growth by facilitating discussions of carefully selected works of visual art. In the Makerspace context, we are inspired by the visual thinking strategy; this is intended to move student learning toward deeper observation and thinking about their “making”. Options may include digital photographs of student learning in action (*the process*) or digital pictures of student artifacts, such as constructed objects (*the product*). Here are some key comprehension skills, connected to the various steps of the VTS discussion model, which may guide you in knowing what to **listen for, pause at, go deeper with, or point out** in student discussion/debriefing:

- **Name It!** …consistently naming the thinking strategies children are using (common language) *“Analyzing is looking for details that can help you understand better.”*
- **Note It!** …consistently noting when children use them (linking student thinking to specific strategies)

“I see… so you were able to outline the plot by putting the photos in order.”
“Amir’s analysis is that the boy in the photo is sad, based on this detail” (points to frown).
“Comparing their outfits, I think one girl comes from a rich family and the other maybe doesn’t.”

…thus, making strategies **explicit not tacit** for both adults and children (metacognition)

Teachers are invited to use three open-ended questions:

1. What's going on in this picture? (describe, explain, formulate, outline, predict, summarize)

2. What do you see that makes you say that? (analyze, compare, contrast, evaluate, infer, support)
3. What more can we find? (all 12 above)

It may be useful to generalize here:

- Six of these comprehension strategies are *somewhat big picture or sequential*
- Six are more *specific or detail-driven*
- All are *essential thinking skills for the design process… and for life!*

Three Facilitation Techniques:

- When paraphrasing, comments remain neutral (all comments are equal; no value/evaluation/good/bad)
- When communicating, point at the area being discussed (i.e. physically gesture)
- Whenever possible, link student comments to strategies (support metacognition by naming and noting)

Students are asked to:

- **Look** carefully at works of art (analyze, infer)
- **Talk** about what they observe (describe, explain)
- **Back up** their ideas with evidence (support, evaluate)
- **Listen to** and **consider** the views of others (compare, contrast)
- Discuss **many possible interpretations** (outline, formulate, predict, summarize, synthesize)

This strategy allowed the learners (student and teacher) to gain new vantage points into learning.

Student Voice

Throughout this process we always put learner voice at the heart of our documentation. We use student voices along with our individual and collective observations and interpretations to create a picture of student learning during Makerspace. This helps us to stay open minded, continue to be curious, and build collective knowledge and understanding. The goal is to interpret the information from our students so as to inform instruction for future Makerspace sessions. By listening and learning from our students, we promote a culture of student empowerment.

We believe that creativity, communication, collaboration, and critical thinking are the tenets of deep learning. It is our belief that engagement in innovative practices such as Inquiry, Design Thinking processes, Makerspaces, and Instant Challenges inspire these practices. We are uniquely positioned in our inner city school to reach our ethnically diverse school community and impact all students with engaging, hands-on making for problem solving and deep learning. In all our planning, our goal is to always place student voice at the heart of all inquiry and innovation.

Our project, Bulletin Board Exploration, highlights one classroom's journey in trying to understand the 4Cs more deeply. When we asked our students about their learning here is what they had to say:

Neha said, "I learned about collaboration while I worked with my team to make our poster!" Aiden offered, "I learned by looking at real pictures and looking closely for examples of the 4Cs. That helped me have a better picture in my mind of what they are." Subhan learned about himself as a learner, "I learned that I can learn better by drawing pictures of the Cs first from my own mind." Keandra reflected, "The poster showed me the 4Cs. When I saw the pictures and all the words underneath it really started to make sense. It was when I saw all the 4Cs together on one board that I really understood."

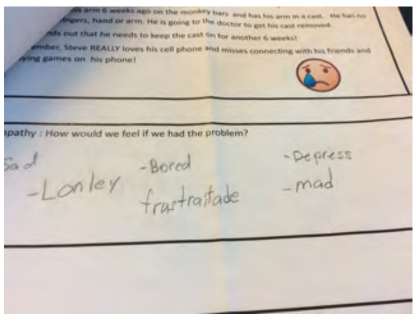
Kathy, our colleague and Grade 3 teacher, and and I believe that our students learned by doing, learned by looking closely and attending carefully to visual cues and details. By honouring student voice and creating collective knowledge and understanding, we felt that our students were able to make deeper connections in their learning. As we all continue to grow and evolve in our understanding of the richness for learning opportunities in Makerspace and Instant Challenges, we will continue to reflect, debrief and create common language for talking and thinking about our learning.

In our exploration of cardboard, (thanks Krissy Venosdale for offering up her term "*carboardification*" and to Caine from Caine's Arcade) our Grade 1 class showed us the beauty of student design. My favourite picture is of this student's "Wish Machine" that is closed in this picture (see the tape over the hole? That is where your wishes go in and then he makes them come true!). It reminds us that makerspace celebrates the process, not the product, in the project.

The Sounds of Repurposing

Room 11 highlights the connection to our environment as the students were invited to explore sustainable practices with cardboard in their Makerspace time.

Room 18, our Grade 5 class, explored empathy in design thinking as they created an iphone holder for a friend who broke his arm!



Teacher Engagement

As professionals, we engage in reflective practice. We think critically about our practices and are engaged in continuous learning and improvement. We believe that by empowering teachers to engage in reflective practice strategies and having a forum to discuss and collaborate in Communities of Practice, we are providing rich opportunities for improvements in student learning and teacher development.

The teachers involved in this journey are engaged in a collaborative planning process with the Inquiry Support Teacher before, during, and after the lesson with a specific focus on learning strategies. We use Google docs for education as a communication tool to support the the work we do. We have created a spirit of collegiality and trust among us that focuses on our shared passion for student and professional learning. We are currently engaged in learning about teacher

(Edublogs) and student blogging (Seesaw) to support the process of learning through reflective practice.

Our colleague, Kathy, describes how she “got into the Maker movement” with her classroom in her blog post titled: The Journey Begins. She concludes:

> The point in which I began to feel like “Yes, this is really working.” was when I noticed a change. About two months into our Makerspace time, the conversations students were having with each other shifted. In the beginning, when they were exploring the materials, their conversations were very much like social free time conversations. During our debrief time, they would summarize what they did, what they made and say that our time was “good” or “fun”. As we continued each session with a debrief, we as facilitators really connected what they were saying back to the 4Cs. I noticed that their conversations began to change. Learners took on different roles in groups where they were creating. Sometimes they would be teacher, sometimes coach, sometimes more like cheerleader. The conversations didn’t wander to a social type of gathering anymore. The focus became clear for them “We have something we want to create and how are we going to make it happen?” It has been a great learning journey for me and my classroom! One that I hope we can build on for the future of all our learners!

In my role as Inquiry Support Teacher, I reflected on my journey with Documentation in Makerspace. In my journal I wrote that:

> Makerspace is full of opportunities for rich and meaningful learning through doing opportunities and being actively engaged in tinkering with the materials. As the students in Room 21 reflected after making their learning in Makerspace and Instant Challenges visible, “We learned by doing!” Moreover, the work that I have been doing in Makerspace, Design Thinking Process, Instant Challenges, and the 4Cs has brought joy! Great joy in doing good work together! Whenever I talk to professional groups I get a little misty eyed when trying to convey the great joy and gratitude I feel everyday when I come to work! I had no idea that, after 25 years of teaching, I would continue to be inspired and challenged everyday in one of the most rewarding jobs in the world! Doing this work with all learners, students and teachers, I have reignited a passion and curiosity for learning and teaching that had been hiding inside me!

Impact on Other Learning Communities

École Victoria-Albert School pioneered Makerspace in Winnipeg School Division. The library has become a flagship for Makerspace both within Winnipeg School Division, and the province of Manitoba. Visitors from across the metro divisions and rural locations frequently visit to observe the physical space, to watch makerspace in action, and to experience the joy of student design, creation, and learning for the future. This past fall, our learning commons was invited to be part of the Manitoba School Library Association tour. We had over 40 teachers from all over

Manitoba visit our space to see the unique and innovative practices that are happening. Our Makerspace was highlighted in a newsletter published by the Enrichment and Innovation consultant, @slcornwell. To see the exciting learning happening at the École Victoria-Albert Makerspace visit us on Twitter @reneesvamakers.

Starting Up

Initially, we received materials from Winnipeg School Division such as LEGO story starters, Rig-a-ma-Jig, DUPLO, BRIO and Lincoln Logs.. They provided us with many of the building materials that helped us start up our program. Further to that, our school was the recipient of the Chapter's Indigo Love of Reading grant award of $100,000.00 in 2014. We allocated some of the funds to support our Makerspace program. The school technology department provided four mini iPads that are used for literacy and numeracy connections like computational thinking. In 2017, we received funding from WSD through a Special Projects Grant called: Deep Learning Through an Innovative Framework. This funding provided both the resources to allow for release time for teachers who are inspired to learn about innovative practice such as blogging, coding, and Google docs for education, as well as opportunities for us to continue to learn, connect and share.

Next Steps

As a next step, we would like to infuse our Makerspace with more "high tech". We see a natural progression to engaging and empowering work with coding, circuitry, green screen, 3D printers, Ipads, Bloxels, and Arduino to name a few. We also envision adding an interactive LEGO wall for team LEGO building projects and adding more sewing machines and tools for building with cardboard. These things would go a long way to providing the students and community that we serve an opportunity to be part of this dynamic and growing movement.

It is All About the Makerspace Mindset

This was our journey but the truth is that you can easily start with whatever you have on hand. Visit ArtsJunktionMB and collect amazing supplies that have been donated by large corporations, dig through the school basement or attic...you might be surprised at what you find has been discarded! If you love to knit, then bring in some knitting needles at noon hour one day! I think the passion and curiosity you have for creating a maker mindset will be the driving force behind your success, not shelves full of stuff that you aren't sure how to use.

In the words of Shauna Cornwell, innovator and leader:

> "Makerspace is not about a "space" and it is not about "maker materials": it is about a "making mindset". A focus on 'making" pushes past the traditional structure of student as consumer of information. It is a culture focused on student as creator. It is about ideas. It is about the joy and exhilaration of putting something new into the world and the rich learning that goes

with the experience of doing so."

Awards and Acknowledgments

In June of 2017 we were the very fortunate recipients of the Ken Spencer Award for Innovation in Teaching and Learning. The Canadian Education Association news release said:

École Victoria-Albert Wins National Innovation Award

École Victoria-Albert School Makerspace transformed its library into a Learning Commons Makerspace, which has provided meaningful hands-on learning opportunities for the majority English-as-an additional-language student population in this ethnically diverse school community. Students work independently and plan their own learning using a variety of Maker materials, from cardboard to Lego robotics for engaging opportunities to think critically and creatively about their given challenge. Teacher debriefings after every Makerspace activity represent a crucial component for their own continuous learning and improvement. By listening to and learning from their students, teachers are providing relevant opportunities for problem solving and deep learning, which is equitable and empowering for all.

WSD news (for the whole article) also acknowledged the Makerspace:

Victoria-Albert Wins National Innovation Award

The school has transformed its library into a Learning Commons Makerspace, which has provided meaningful hands-on learning opportunities for the majority of its students, including a significant English-as-an additional-language student population.

Students work independently and plan their own learning using a variety of Maker materials, from cardboard to Lego robotics for engaging opportunities to think critically and creatively about their given challenge. Teacher debriefings after every Makerspace activity represent a crucial component for their own continuous learning and improvement.

Our local newspaper, *Canstar* featured us in an article titled, Taking Learning to the Next Level.

"Every time we meet with a group…we learn as much from them, their questions and their passions, as we do as a group who meets to explore questions of understanding and learning and next steps," she said. "It's the students who make it all worthwhile."

École Victoria-Albert School was the first official Makerspace in the Winnipeg School Division, and their legacy has encouraged many schools in the province to use the same system. At the ceremony, Doug Edmond, WSD's director of research, planning and systems management, said 30 schools in the division run a Makerspace program.

Canstar Community News *(for the whole article):*

Schoolgirls Rule Engineering Contest

By: Ligia Braidotti

04/17/2017

2017 Fluor Engineering Challenge Grant Drawing April 3, 2017 (Winnipeg, MB) – Students at École Victoria-Albert School are one of ten proud winners of the 2017 Fluor Engineering Challenge grants. Working with limited materials and their imaginations, the students used the engineering design process to build an efficient irrigation design using only cups, tin foil, tape and craft sticks. "All the girls agreed that their favorite part was working together to come up with ideas and choosing one that they liked the best," said Renee Sanguin, Inquiry Support teacher at École Victoria Albert School. "They loved that they all got along and never fought about how to build it."

In Conclusion

While being recognized for our commitment to innovation and inquiry was an amazing part of our journey, I would have to say that the biggest success for me has been the joy in student's faces when they are being challenged to work together to be creative problem solvers. How the students are learning and taking this mindset out into the world is the real reward. We can all be makers and creators; we don't need a lab or a school to do it! We need a mindset that allows us to take chances and find the solutions to our problems.

The Makerspace program at École Victoria Albert School promotes access to learning which is deep, equitable and empowering for all. It opens doors for learners in a very personalized and experiential learning environment. The foundational tenets of deep learning driven by Makerspace has transformed our school from a traditional teacher directed model to one where students are at the heart of all planning and learning. Our focus is on learning skills that will take our students into the future of change that is their reality.

As we live in a dynamic, rapidly changing world, educators must strive to meet the unique and unprecedented learning needs of our students in the midst of these changing times. In 2015, The Economist Intelligence Unit (the world leader in global business intelligence) completed a study focusing on preparing our students for the future and what skills that reality will demand. After surveying respondents from countless industries, business sectors and fields of education from countries around the world, the study showed that organizations felt the top five critical skills for employees today are: problem solving, teamwork, communication, critical thinking and creativity. The École Victoria Albert Makerspace initiative is an example of the transformative change needed in the public education system to prepare our learners with the 21st Century literacies, skills, competencies and attitudes necessary for today's world. The École Victoria Albert Makerspace develops 21st Century learners as they become creators, critical thinkers, problem solvers, collaborators, communicators, and most importantly, life-long learners.

Renee Sanguin, the Inquiry and Innovation Support teacher at École Victoria-Albert School facilitates learning in the library learning commons and in classrooms. Renee is called upon regularly to facilitate adult learning sessions and workshops for teachers within Winnipeg School

Division and beyond. Her knowledge and expertise in the areas of competency based learning, design thinking, reflective practice and the art of student driven creation, enlighten and inspire many! You can follow her on Twitter @reneesvicalmakers

Renee Sanguin
Inquiry Support Teacher
École Victoria-Albert School
110 Ellen Street
Winnipeg, Manitoba
R3A 1A1
204-943-3459

Chapter 4 - We Are All Makers: A First Grade Inquiry Unit

by Amanda Ross

Introduction

Our division's focus is on "deeper learning" and I am part of the Deeper Learning Team at my school. We have eight competencies in our learner circle: the 6 C's (Critical Thinking, Communication, Collaboration, Citizenship, Character, and Creativity) and 2 L's (Literate and Learner). We also focus on four design elements when building learning experiences:

- Learning Partnerships,
- Leveraging Digital,
- Pedagogical Practices, and
- Learning Environments.

I want my teaching to be more student-led, so this "unit plan" is a flexible plan, with the intention that student interest and questions might lead us in a new direction and I am ok with that. The big ideas that I want students to explore during this unit are learning about different materials and their characteristics, exploring how materials can be joined together to create objects, and using a simple design process idea (like the TMI Robot from Invent to Learn). I would also like to explore hobbies and jobs that involve making so that students can see that it is an important skill to work on and get better at. This is our first inquiry unit of the year, so it will be more teacher-guided to begin with. As we progress throughout the year, I will gradually release more responsibility to the students and suggest inquiry units that are based on student interest, current events, or other issues that come up in our classroom.

Unit Plan Design

Curricular Areas:	Science	Cluster 3: Characteristics of Objects and Materials Cluster 0: Scientific Inquiry and Design Process Skills
	ELA	Practice 3 - Language as Exploration and Design ● Research and study topics and ideas ● Interpret and integrate information and ideas from multiple texts and sources ● Manage information and ideas ● Invent, take risks, and reflect to create possibilities
	Social Studies	Skills: Active Democratic Citizenship (collaboration), Managing Information and Ideas, Critical and Creative Thinking, and Communication
Design Elements:	Learning Partnerships	● walking field trip to M&R Furniture (builds real wood furniture there) ● bring in some "experts" (parents in some related field--construction, sewing, etc.)
	Leveraging Digital	● use SeeSaw digital portfolio to document learning and make reflections ● share our learning with others using Twitter ● use various apps to create parts of the project ● do research with ebooks and videos
	Pedagogical Practices	● encourage students to come up with the questions we want to answer ● give students choice in how they complete parts of the project ● setting out provocations for building toys/art materials
	Learning Environments	● whole group meeting area for direct instruction mini lessons, and reflecting on what we've learnt or done so far ● many different areas for small group collaboration ● MakerSpace area and Tinkering Table for exploring different materials and how to design and create things

		• access to large classroom library and five classroom iPads • access to school library for research and computer lab or laptop carts
Competencies:	Critical Thinking	• choosing a problem for final project and coming up with possible solutions • revising creations/products when they don't work out • coming up with multiple solutions to problems/ tasks • evaluation which materials are better suited to certain tasks
	Communication	• asking questions to "experts" or guest speakers • sharing and listening during reflection time • presenting their final project
	Collaboration	• instant STEM challenges in groups • group projects (materials poster, sorting activities, etc.) • asking classmates for help or ideas when working on a project (ask 3 before me)
	Citizenship	• show care and responsibility for the environment • make good choices during work time
	Character	• respecting other people's thoughts and ideas • showing compassion for animals' needs in final project
	Creativity	• explore and find new ways to create using the building toys and art materials • use innovation to create a new object for their final product
	Literate	• have a purpose for reading and writing • research and find information on a particular topic • shared writing experiences for labels/reflections on the project wall
	Learner	• learn how to ask meaningful questions and how to find the answers • learn where to find information

		• see how the topic relates to real-life (parents' jobs, hobbies, interests, etc.) • make connections between topics and subject areas

Unit Plan Outline

Main Content Area	Grade 1 Science, Cluster 3: Characteristics of Objects and Materials
Timeline	September (about 3-4 weeks), roughly about one hour a day
Driving Question	How are the objects around us made and who makes them?
Need to Knows*	• What is the difference between objects and materials? • What are the different characteristics of materials? • How can materials be joined together? • Which materials are best for a particular job? • Who makes objects? • What is the process for making things? • What objects can we make? • What can we use to make objects? • Do we need to have a purpose for making something? • How can we think about the environment as we are making?
Student Projects	• Digital class book about objects around the school • Posters about materials and their characteristics • Design project and presentation
Step 1: Looking and Noticing (Provoke)	• Activate prior knowledge--what have they ever made, what do they know about building or making things, what do their parents make • Discuss students' experiences with building or making--is it easy or hard? do you think about ideas before you make? where do you get ideas from? • Create a list of "need to know" questions* (teacher might lead students into some of these, but will accept all of their ideas) • Explore the MakerSpace area (what materials are available? what can we do here? what are you interested in making?)
Step 2: Developing the Project and Collecting Data (Explore)	• Use books and videos to learn about the characteristics of different materials; create posters about different materials describing their characteristics and giving examples of objects made of them • Have groups look at a variety of materials and decide on a way

	to sort them based on some sort of characteristics (hard/soft, bendy/rigid, etc.); have them do a gallery walk and explain how they sorted the materials ● Give each group a collection of materials, present a problem (ex. you need to make a mitten out of one of these materials), and have students discuss which material would be suited and why ● Explore building toys (Lego, K'nex, Lincoln Logs, Locktagons, Tinker Toys, Bristle Blocks, Keva Planks, etc.) to learn about different joining and building techniques ● Explore art materials (plasticine, paint, loose parts, Wikki Stix, play-dough, etc.) to make artwork ● Explore the Tinkering Table to see how objects are made by taking them apart ● Go outside and use natural materials (rocks, sticks, leaves, etc.) to make something (but still being respectful of the environment) ● Complete some instant STEM challenges to practice engineering skills and joining techniques ● Brainstorm jobs/hobbies that require making or building ● Go on a scavenger hunt around the school looking for objects that are made up of a variety of materials; create labelled images on PicCollage Kids to create a printable or ebook about materials ● Explore the idea of reusing and recycling and make a plan how to be environmentally friendly when making and creating ● Go on a walking field trip to M&R Furniture to ask questions to an "expert" (carpenter) ● Invite parents or other guests in to talk about how they are makers ● Final Project: Design an object that will help an animal somehow using materials found in the classroom or home. (Brainstorm things animals might need help with--getting food, somewhere to sleep, something to play with, etc.). Encourage the TMI model--think, make, improve.
Step 3: Concluding the Project (Reflect)	● Presenting the final project: students can create a video, poster, or oral presentation (or another option if they can think of one) to explain what they created and why, the process it took to create, difficulties and successes they had, and explain a bit about the materials they used and joining techniques they used. ● Reflecting on the entire collaborative project wall. Go back to our list of need to know questions to see if we answered everything. See if there are any further questions they want to explore.

Formative Assessment	● Sharing samples of work on SeeSaw for parents (and teacher) to see--this can be student chosen as well. They can reflect on what they were thinking at the time or why they chose to include that learning piece. ● Conferences with students as they research and work on aspects of the project. ● Observing students while they are working and taking anecdotal notes/pictures. ● Reflection time at the end of each class: what did we discover or learn today? What are we going to continue on tomorrow? Use pictures of students to discuss the learning that we see happening.
Summative Assessment	● Collaborative project wall: we will have a bulletin board designated for our inquiry unit. We will keep track of all of our learning: our list of need to know questions (and our answers), pictures of our creations/STEM challenges, materials posters, etc. ● Final design project presentation

Credits

Deeper Learning image from hsd.ca

Amanda Ross is a first grade teacher in the Hanover School Division. She has been teaching for eleven years, the past 7 years have been in first grade. She is currently working on her Post Bacc in Education at the University of Manitoba. She loves incorporating new ideas and technologies into the classroom. When she's not with her first graders, she is spending time with her two munchkins at home. You can find her on Twitter at @mrsrossgrade1 or over at her blog First Grade Garden. You can also contact her by email at amross@hsd.ca.

Chapter 5 - Early Years Makerspace Resource List

Compiled by Amanda Ross

Table of Contents

All book and supply images are taken from Amazon.ca (unless otherwise noted)
If you click on any of the images it will take you to Amazon.ca (or other link)
Prices listed are subject to change--check the given links for current prices!

Introduction

What is a MakerSpace?

mak·er·space (□mākər□spās/ *noun*)

1. a place in which people with shared interests, especially in computing or technology, can gather to work on projects while sharing ideas, equipment, and knowledge.

Definition from here

About This Resource List

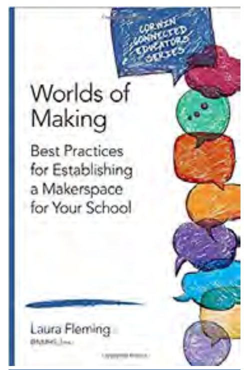
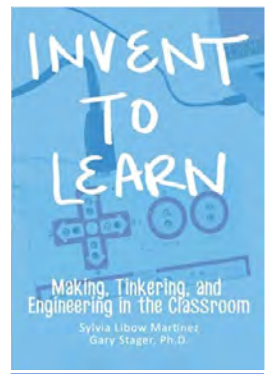
This resource list is for any early years teacher (K-4) who is interested in setting up a MakerSpace in their classroom or school, who wants to learn more about the MakerSpace mentality or STEM/STEAM education, or who just wants some ideas on new things to try out in their classroom.

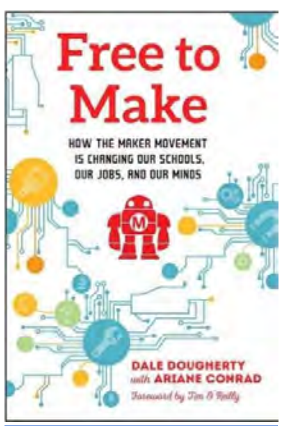
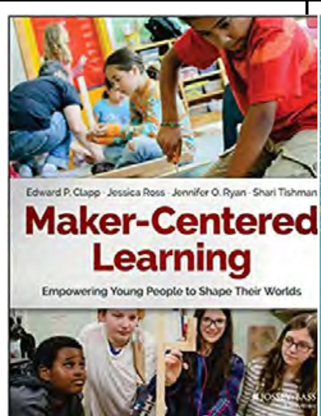
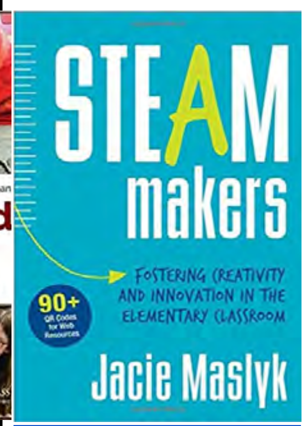
There is a big push in education right now to teach students 21st century skills, such as collaboration, creativity, and communication. A lot of these skills can be taught using the MakerSpace mentality.

While searching for different ways to incorporate the “Maker” mentality into my first grade classroom this year, I came across so many amazing resources that I wanted to share with other teachers. I have included a list of my favourite maker-type picture books, some building/construction toys that are amazing for young learners, and lots of technology ideas teaching coding, robotics, engineering, and more. You will also find some helpful Twitter accounts to follow, blogs to check out, or teacher resource books to give you more information about MakerSpaces.

Book Ideas for Teachers

Here are some good professional development books to get you started in planning a MakerSpace or just to learn more about what it is. Another fantastic place to go for PD is Twitter. These are some good people to follow for MakerSpace ideas: @gravescolleen (co-author of "The Big Book of Maker Space Projects"), @Makerspaces_com, @LFlemingEDU (author of "Worlds of Making"), @DianaLRendina (blogger at renovatedlearning.com), @MakerEdOrg, @WSDSteam, @GwynethJones (blogger at www.thedaringlibrarian.com), @dsmacdonald (see her MakerSpace), @spencerideas (author of "Launch"), @thenerdyteacher (author of "Your Starter Guide to MakerSpaces"), @krissyvenosdale (and her space @KLSlaunchpad)

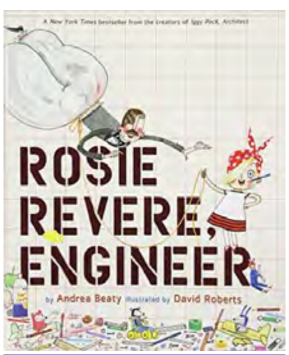
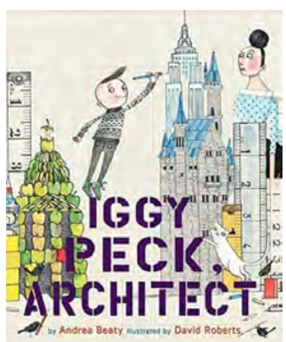
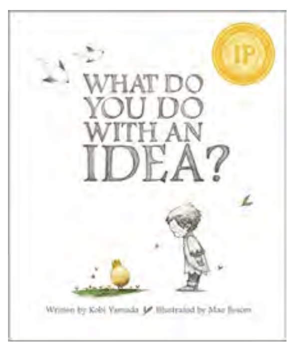
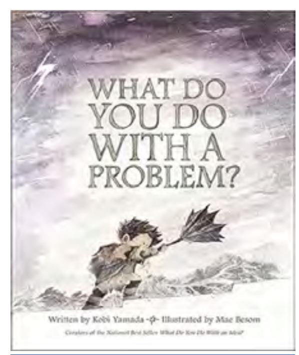
 The Big Book of Maker Space Projects by Colleen Graves and Aaron Graves ● Lots of amazing project ideas with simple step-by-step directions. ● $23.79	 Your Starter Guide to Maker Spaces by Nicholas Provenzano ● An easy read that goes into what is Making, why is it important, and how we can get started. ● $27.45	 Worlds of Making: Best Practices for Establishing a Makerspace for Your School by Laura Fleming ● $16.35 ● Coming out with a new book about making great MakerSpaces (Fall 2017).	 Invent to Learn: Making, Tinkering, and Engineering in the Classroom by Sylvia Libow Martinez and Gary Stager ● Lots of theory and history about the Making movements as well as some practical ideas about equipment and supplies. ● $36.31

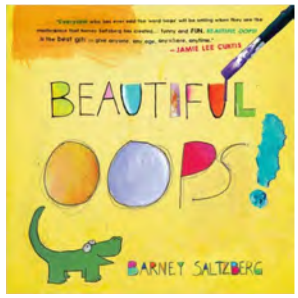
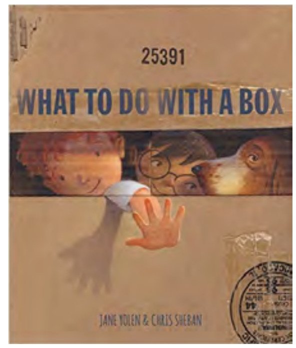
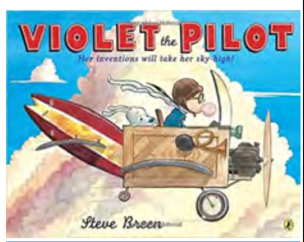
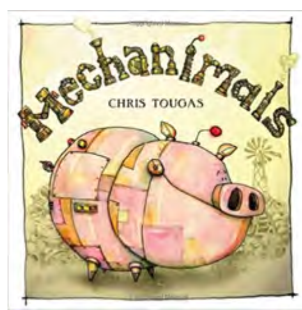
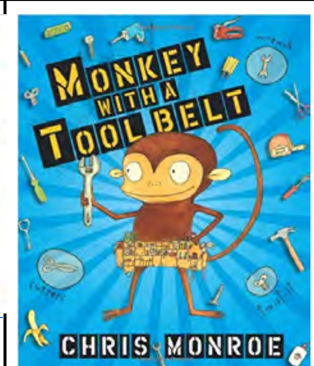
 Launch: Using Design Thinking to Boost Creativity and Bring Out the Maker in Every Student by John Spencer & A.J. Juliani ● How to incorporate design thinking or the creative process into the classroom. Talks about project based learning a bit. ● $32.04	 Start Making!: A Guide to Engaging Young People in Maker Activities by Danielle Martin ● How to plan maker activities and projects at home or school. ● $24.79	 Tinkering: Kids Learn by Making Stuff by Curt Gabrielson ● How to incorporate tinkering into your classroom or MakerSpace with some project ideas. ● $26.44	 Makerspaces by Samantha Roslund and Emily Puckett Rodgers ● A look at what are Makers and MakerSpaces. ● $19.29
 Makerspaces: A Practical Guide for Librarians by John J. Burke ● This guide helps you set up your	 Free to Make: How the Maker Movement is Changing Our Schools, Our Jobs, and Our Minds by Dale Dougherty	 Maker-Centered Learning: Empowering Young People to Shape Their Worlds by Edward P.	 Steam Makers: Fostering Creativity and Innovation in the Elementary Classroom by Jacie Maslyk

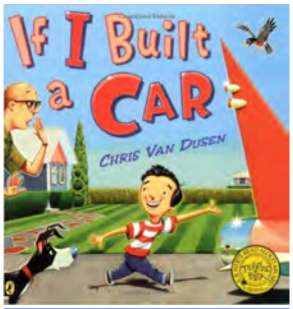
MakerSpace--where to get supplies and some project ideas to get you started. ● $80.03	● A look into the Maker movement and how it is going to change our schools. ● $18.45	Clapp, Jessica Ross, Jennifer O. Ryan, & Shari Tishman ● Looks at the practices of maker-centred learning and explores resources and routines for fostering making in your classroom or school. ● $34.49	● How to integrate STEAM and Making into your everyday classroom practice. ● $45.08

Book Ideas for Students

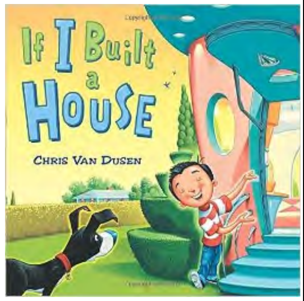
This is a list of picture books for inspiring students to make mistakes, dream big, create, invent, solve problems, believe in themselves, explore, and MAKE.

 Rosie Revere, Engineer by Andrea Beaty ● Rosie is a brilliant inventor and wants to create a flying machine! Her great-great aunt Rose teaches her to look at her "failures" in a different way. ● $17.06	 Iggy Peck, Architect by Andrea Beaty ● Iggy Peck's passion is building. He sometimes uses interesting and bizarre building material choices. After his teacher declares she doesn't like architecture, Iggy has a challenge to	 What Do You Do with an Idea? by Kobi Yamada ● A super sweet book about how to take and nurture your ideas. ● $22.49	 What Do You Do with a Problem? by Kobi Yamada ● How to look at your problems and think creatively and critically about how to solve them. ● $22.71

	face! ● $19.92		
 The Most Magnificent Thing by Ashley Spires ● The girl wants to build something magnificent but faces many challenges and eventually wants to give up. A good lesson on perseverance. ● $14.45	 Beautiful Oops! by Barney Saltzberg ● The illustrator takes splotches, spills, rips, and other mistakes and turns them into something beautiful! ● $20.66	 What To Do With a Box by Jane Yolen ● A beautiful story about how children can use their information to transform a box into anything. ● $22.90	 Anything Is Possible by Giulia Belloni ● Sheep is a dreamer and wolf is practical, but together they collaborate and work together to build a flying machine! ● $15.78
 Awesome Dawson by Chris Gall ● Dawson collects junk and turns them into amazing machines. ● $20.16	 Violet the Pilot by Steve Breen ● Violet is a talented inventor who wants to earn the respect of her classmates. ● $11.87	 Mechanimals by Chris Tougas ● A farmer builds his own animals from scrap metal left on his farm. ● $9.85	 Monkey with a Tool Belt by Chris Monroe ● Chico Bon Bon can fix or make anything and shows how to creatively solve a problem with the right tools! ● $22.87



If I Built a Car
by Chris Van Dusen
- A great book about imagination and all the possibilities out there as Jack dreams about and builds his fantasy car.
- $9.89



If I Built a House
by Chris Van Dusen
- Similiar to the car book, Jack dreams about his ideal house and sets out to build it!
- $19.19

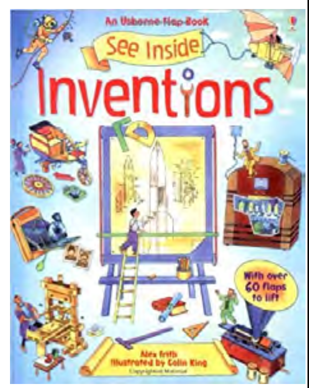
In Mary's Garden
by Tina & Carson Kugler
- Based on real life artist Mary Nohl who used common objects to make beautiful art. Her garden became her gallery.
- $19.60

An Illustrated Timeline of Inventions and Inventors
by Kremena T. Spengler
- A look at inventions of the past in a neat, illustrated timeline.
- $7.95

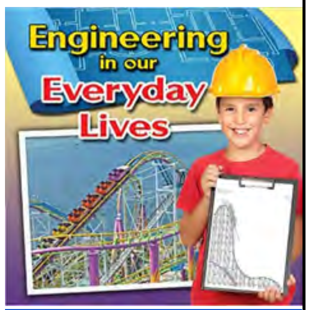
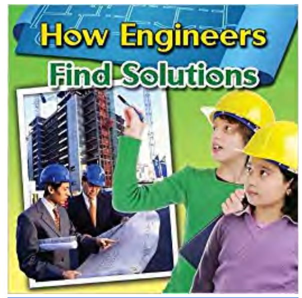
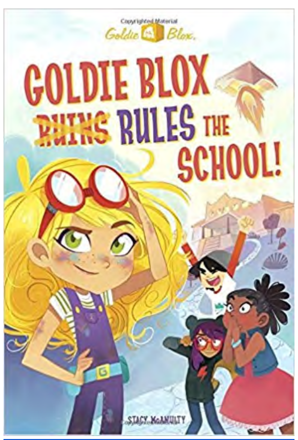
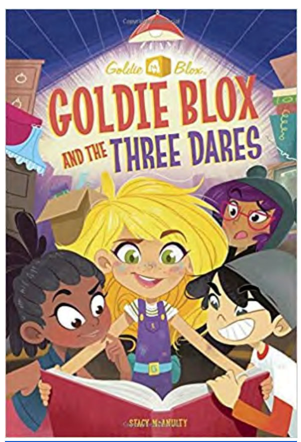
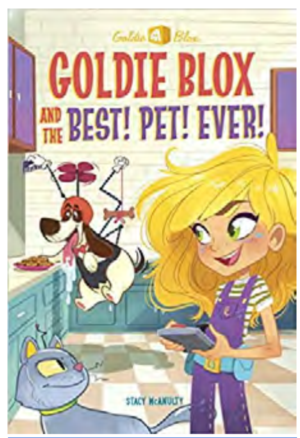
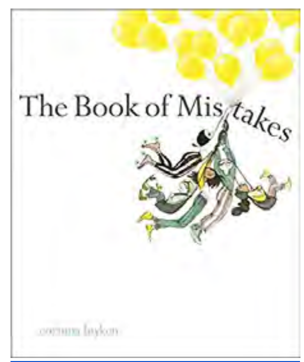
The Marvelous Thing That Came from a Spring: The Accidental Invention of the Toy That Swept the Nation
by Gilbert Ford
- A cute story about the accidental creation of the super famous toy.

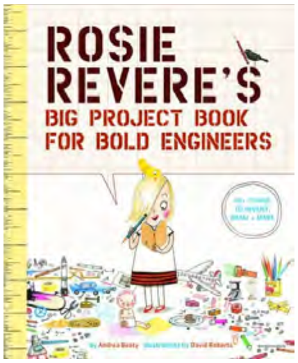
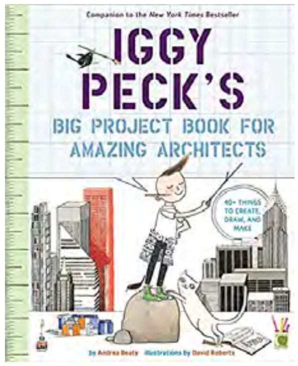
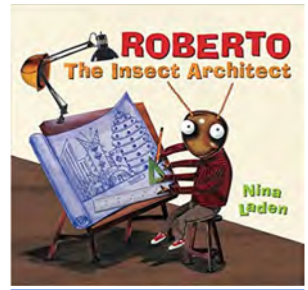
Papa's Mechanical Fish
by Candace Fleming
- Based on real-life inventor Lodner Phillips, who invents a submarine and takes his family for an underwater ride in Lake Michigan.
- $19.74

Oh, the Things They Invented!: All About Great Inventors
by Bonnie Worth
- The Cat in the Hat goes back in time to meet with different inventors.
- $10.88



See Inside Inventions
by Alex Frith
- A lift the flap book that looks at important inventions throughout history.
- $17.95

• $21.59			
 Engineers Solve Problems by Reagan Miller • An easy reader series that looks at the roles of engineers. • $10.79	 Engineering in Our Everyday Lives by Reagan Miller • An easy reader series that looks at the roles of engineers. • $10.79	 How Engineers Find Solutions by Robin Johnson • An easy reader series that looks at the roles of engineers. • $10.79	 Engineers Build Models by Reagan Miller • An easy reader series that looks at the roles of engineers. • $10.79
 Goldie Blox Rules the School! by Stacy McAnulty • Early chapter book (ages 6-9) • A series to go with the Goldie Blox toys. • Encourages girls to become builders and engineers. • $9.41	 Goldie Blox and the Three Dares by Stacy McAnulty • Early chapter book (ages 6-9) • Encourages girls to become builders and engineers. • $9.12	 Goldie Blox and the Best! Pet! Ever! by Stacy McAnulty • Early chapter book (ages 6-9) • Encourages girls to become builders and engineers. • $9.50	 The Book of Mistakes by Corinna Luyken • Another book that takes spills, splotches, and mistakes and turns them into something beautiful. • $24.74

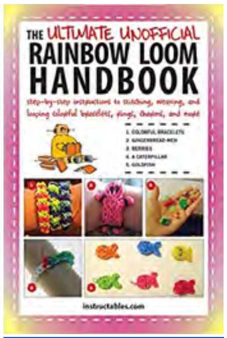
			
Rosie Revere's Big Project Book for Bold Engineers by Andrea Beaty ● A project idea companion to the Rosie Revere book. ● $17.48	Iggy Peck's Big Project Book for Amazing Architects by Andrea Beaty ● A project idea companion to the Iggy Peck book. ● $15.85	National Geographic Readers: Robots by Melissa Stewart ● A nonfiction book packed with pictures about robots. ● $4.49	Roberto: The Insect Architect by Nina Laden ● Roberto follows his dream of being an architect, despite his wood-eating family's discouragement. ● $10.88

MakerSpace Supplies

MakerSpaces don't have to be stocked with the most expensive technologies or tools. If you are just starting off, get donations of cardboard and various craft supplies to get you started! If you have a budget and want to start exposing your students to different ways of "making", here are some interesting, fun tools to give a try in your MakerSpace area!

			
Kraft Tape Dispenser with 10 Assorted Tape Rolls	Crop-a-Dile ● Punches holes in anything--cardboard, popsicle sticks, etc.	[1]Pyssla Beads ● Similiar to perler beads; make a	[1]Pyssla Shape Set ● Students can create different patterns, shapes, and designs on the different boards.

• $53.16 • Tape refills: $41.27	• $36.86	design and heat with an iron. • $6.99	• $2.99
 Tape It & Make It: 101 Duct Tape Activities by Richela Fabian Morgan • Lots of project ideas of things to do with duct tape! • $15.06	 A Kid's Guide to Awesome Duct Tape Projects by Instructables.com • More duct tape ideas. • $22.72	 Fiskars Duck Edition Scissors • Cuts duct tape without leaving residue on the blades. • 5” $10.99 • 8” $12.99	 [2]24 Roll Variety Pack Solid Colors of All Purpose Duct Tape • $28.77 + $15.56 Shipping (US)
 Makey Makey • Turn anything into a keyboard. • $71.95	 3Doodler Start Essentials Pen Set • Kid-safe 3D doodling pen. • $72.99	 3Doodler Refills • Regular colours: $16.99 • Neon colours: $33.13	 [3]Education Makedo™ Kit • Awesome cardboard engineering kit. • Safe tools for cutting cardboard, plastic screws and screw drivers for connecting cardboard together. • Set for 12-24 makers. • $159.95 • Beginner Kit:

			$15.95
 The Ultimate Unofficial Rainbow Loom Handbook by Instructables.com ● Project idea for things to create with rainbow looms. ● $18.81	 Rainbow Loom Bands with Metal Hook ● Students can create amazing creations with these looms. ● $18.77	 Rainbow Loom Bands 9600 Pieces 16 Colors ● Huge refill pack of colourful elastic loom bands. ● $26.99	 Rainbow Loom Finger Loom Party Pack ● Other tools for rainbow loom creations. ● $17.00
 Melissa & Doug Wooden Multi-Craft Weaving Loom ● A large loom for weaving. ● $32.95	 MindWare Pottery Wheel Game ● A beginner’s pottery wheel. ● $99.95	 Amaco Air Dry Modeling Clay, 10-Pound, White ● Air dry clay to use with the pottery wheel. ● $11.15	 Maker Lab: 28 Super Cool Projects: Build * Invent * Create * Discover by Jack Challoner ● Another great project ideas book. ● $21.73

 Kids Leather Work Gloves ● Perfect for a "tinkering" station. ● $7.99	 Learning Resources Safety Goggles ● Good for science experiments and tinkering. ● $5.00	 Apollo Precision Tools Stubby Set, 4-Piece ● Smaller size tools for smaller hands. ● $23.99	 [2]Stanley Jr. Children's 5 Piece Toolset ● $13.00 + $14.24 Shipping (US)

[1]*Images from IKEA.ca*
[2]*Images from Amazon.com*
[3]*Image from Wintergreen*

Common School Supplies

- tape (regular, masking, duct, electrical, washi)
- glue (sticks, bottles, hot glue)
- paper (white, lined, graph, construction, cardstock, newspaper, tissue)
- brads, paper clips, binder clips, binder rings
- staplers and staples
- hole punches (regular, three punch, shape punches)
- crayons, pencil crayons, markers
- paints (tempera, acrylic, water colour) and brushes
- plasticine, air dry clay, play-doh

Dollar Store Stuff

- popsicle sticks
- pipecleaners
- plastic cups
- straws
- paper bags
- Q-tips

- pom poms
- velcro
- tin foil
- balloons
- coffee filters
- tooth picks
- ribbon
- stickers
- foam
- felt
- cotton balls
- plastic utensils

Ask for Donations

- egg cartons
- milk cartons
- cardboard
- bubble wrap
- yogurt/fruit cup containers
- newspaper
- yarn, ribbon
- spools
- toilet paper/paper towel rolls
- buttons
- corks
- plastic lids
- scrap material
- nuts and bolts
- old keys and/or locks
- old electronics
- outdoor materials (sticks, rocks, shells, acorns)

Another Resource: Arts Junktion MB

http://artsjunktion.mb.ca/

312-B William Ave. (East Side, Loading Dock Entrance), Winnipeg, Manitoba

A place to get free art and making supplies! They also host a bunch of different workshops.

Organization Ideas

There are so many ways to organize MakerSpaces. You could use an extra classroom, a part of the library or science lab, a corner in your classroom, or even a mobile cart that can be wheeled from room to room. Here are some neat ideas for organizing your space and stuff!

[1]Free Supply Labels from Teach Outside the Box
- Free labels to organize your supplies.

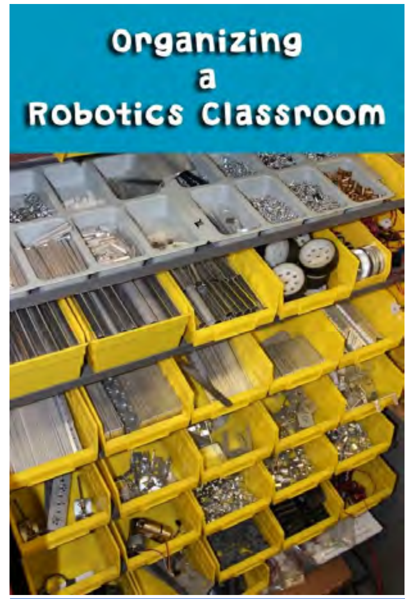
[2]Yarn Organizer
- Use a laundry hamper as an easy yarn dispenser!

[3]Small Classroom MakerSpace Idea
- Neat storage idea: plastic drawer units with a table top on top.

[4]Mini Maker Space Cart for Kids
- Free Supply Checklist

[5]Classroom MakerSpace
- Love the peg board idea for storing supplies.
- Hang scissors and other tools for easy access.

[6]Colour-Coded Clean Up
- Bins are colour-coded into different categories: electronics, attaching, tools, materials, wood, soft stuff, sewing/crafts, paint/sewing.

Download ● Make a cart that can be moved around the room or from room to room.		
 [4]DIY PVC Pipe Tape Dispenser ● Create a scissor and tape station using PVC pipe.	 [7]Introducing our BPC Makerspace ● IKEA shelving makes great storage for a MakerSpace.	 [8]Organizing a Robotics Classroom ● How to set up and organize robotics parts.

[1]*Image from Teach Outside the Box*
[2]*Image from Pinterest*
[3]*Image from Mrs. Byrd's Learning Tree*
[4]*Image from Left Brain Craft Brain*
[5]*Image from Jasztalville*
[6]*Image from It'll be.jessed that easy…*
[7]*Image from STEAMspace*
[8]*Image from Robo Matters*

Teacher Resources

Here are a list of blog posts, TeachersPayTeachers products, and other websites that will provide you with more information about creating and running a MakerSpace.

<table>
<tr>
<td>


[1]Maker Mats
● Suggested use is for a homework idea, but you could hang these up as monthly idea boards.
</td>
<td>


[1]STEM Bins
● Includes task cards for ideas of things to build.
</td>
<td>


[1]Outdoor Discovery STEM Bins
● Includes more task card ideas--outdoor themed.
● Build or create with natural objects (wood cookies, twigs, acorns, rocks, etc.).
</td>
<td>

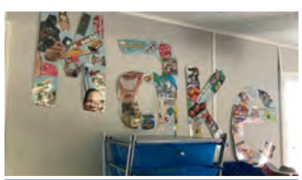
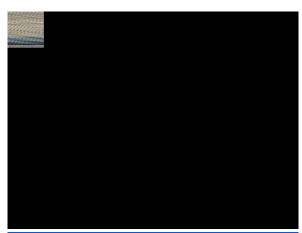

[1]Seasonal Task Cards for STEM Bins
● Seasonal task cards, famous structure task cards, furniture task cards, etc.
</td>
</tr>
<tr>
<td>


[1]Unplugged Coding All Year Bundle
● Hands on activities for young students to practice simple directional coding.
</td>
<td>

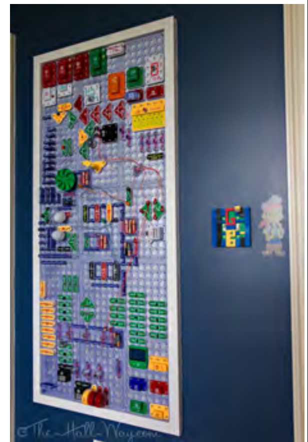

[1]MakerSpace Posters and QR Videos
● Posters to hang up in your MakerSpace with QR code links to videos on how to make different attachments, paper
</td>
<td>


[1]Maker Space FREE Guide to Getting Started
● A free guide to set up a MakerSpace: supply lists, ideas for organization and scheduling, and more
</td>
<td>


[2]Create an Amazing Low-tech Library Makerspace With These Easy Ideas
● Incorporate low-tech ideas into your
</td>
</tr>
</table>

	techniques, etc.		MakerSpace such as cardboard, duct tape, and perler beads.
 [3]114 Tips to Create a STEAM MakerSpace ● Items you need, websites to use, apps, organization ideas, and STEAM units of study to incorporate.	 [4]MakerSpace Starter Kit ● This blog has a ton of amazing ideas for MakerSpaces. ● She has four "corners"--colouring, duct tape creations, Lego, and a Makey Makey Coding Corner.	 [5]Makerspaces Without a Space: Circulating Maker Kits for the School Library ● If you don't have the space in your classroom or library to have an official MakerSpace corner, you can create circulating maker kits that students can bring anywhere to work on.	 [5]Awesome Elementary Makerspace Resources ● A list of people and blogs to follow, best resources, and good Pitnerest boards to check out.
 [6]How to create your own iPad recording booth for the	 [7]How to Start a Makerspace for Less Than $20 in Your School ● How to reuse things from around your school to start up a MakerSpace.	 [8]Early Life Foundations - Walker Approach Facebook Group ● An Australian, preschool play-based/inquiry learning approach.	 [9]Sculpture Center Signage ● Love the glue anchor chart--what

classroom. ● Create a recording booth to block out the background classroom noise when students are recording movies or adding voice clips to their digital portfolio.		● Lots of great provokation ideas. ● Some awesome "tinkering" table ideas.	kind of attachment tool will work best for which job?
 [10]Tapping into Declarative and Procedural Knowledge in the Art Room ● Anchor charts for cardboard attachment techniques.	 [11]Contribute Your Creativity to a Tee by artist Danny Murphy ● Use a pegboard and colored golf tees to create an ever-changing and collaborative mural.	 [12]MakerSpace Cart ● Another portable MakerSpace cart idea. ● Love the design process on the side: imagine, plan, create, improve, share.	 [13]DIY Cardboard Tube Construction Toy ● Cut slits into cardboard tubes for an easy DIY building toy.

 [14]5 Reasons Why Maker Education Can Improve Student Learning ● Good infographic to share with admin and/or parents about the benefits of Maker Education.	 [15]Exploring Coding With Kindergarteners (with and without technology) ● Some fun, hands on intro to coding activities. ● Build a block maze for Sphero.	 [16]Stem Challenges Pinterest Board ● Lots of challenge ideas.	 [17]Snap Circuits Wall ● Create a DIY Snap Circuits wall using the baseplates. (Similar to a Lego wall.)
 [18]Using BeeBots in the Classroom ● Great ideas on how to use the BeeBot in an early year's classroom.	 [19]The Digital Scoop ● Free task cards to use the Dash and Dot robots with young students here and here.	 [20]5 Ideas for Implementing Ozobots ● Ways to integrate the Ozobot with curriculum.	 [21]Ozobot Costumes ● Challenge students to make costumes for the Ozobot to use them in stories, movies, or games.

<table>
<tr>
<td>

[22]Little Bins for Little Hands Blog
• Tons of amazing STEM ideas.</td>
<td>

[23]Using a Green Screen to Support and Enhance Your Curriculum
• Some ideas for creating movies with a green screen.</td>
<td>

[24]Krissy Vensodale Blog
• She is an “innovation coordinator”. Check out her tour of her beautiful “Launch Pad” space here.
• See her list of resources here.
• Download and use her Maker posters here and here.</td>
<td>

[25]Starting a MakerSpace from Scratch by Colleen Graves
• Steps on how to get started on a school MakerSpace.</td>
</tr>
</table>

[1]*Images from TeachersPayTeachers*
[2]*Image from Ideas + Inspiration by Demco*
[3]*Image from Education Closet*
[4]*Image from The Daring Librarian*
[5]*Image from A Wrinkle in Tech*
[6]*Image from blairsmithteaching*
[7]*Image from We Are Teachers*
[8]*Image from Early Life Foundations - Walker Approach Facebook Group*
[9]*Image from Elmwood’s Art Studio*
[10]*Image from The Art of Education*
[11]*Image from Flickr*
[12]*Image from The Library Voice*
[13]*Image from PickleBums*
[14]*Image from Ed Tech With Hoekstra*
[15]*Image from A Day in First Grade*
[16]*Image from Pinterest*
[17]*Image from The Hall Way*
[18]*Image from Della Larsen’s Class*
[19]*Image from The Digital Scoop*
[20]*Image from Talkin’ Pinata*
[21]*Image from Danielle’s Place*
[22]*Image from Little Bins for Little Hands*

[23]*Image from [Mrs. B's First Grade]*
[24]*Image from [Krissy Venosdale]*
[25]*Image from [Edutopia]*

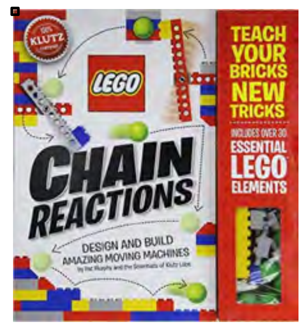
Lego Resources

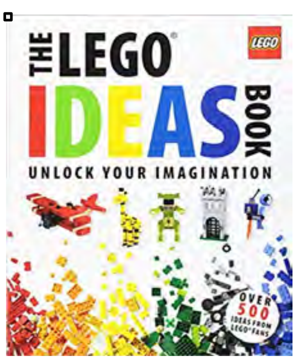
Here are some great resources to get your Lego table, wall, or center set up as part of your MakerSpace! Students (and adults) of any age love to explore, play, build, and create with Lego.

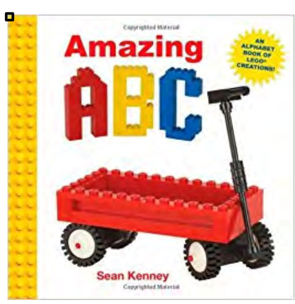
<table>
<tr>
<td>

LEGO Play Book: Ideas to Bring Your Bricks to Life
by Daniel Lipkowitz
● An idea book for Lego creations.
● $21.08</td>
<td>

365 Things to Do with LEGO Bricks
by Simon Hugo
● Something new to do and make with Lego for every day of the year. Includes a built in timer/activity selector.
● $27.09</td>
<td>

LEGO Awesome Ideas
by Daniel Lipkowitz
● What amazing things can you build with Lego?
● $21.24</td>
<td>

LEGO Chain Reactions: Make Amazing Moving Machines Toy
by Klutz
● A neat book with unique Lego pieces for making tricks, spins, swings, lifts, etc.
● $25.26</td>
</tr>
<tr>
<td>

</td>
<td>

Amazing ABC: An</td>
<td></td>
<td>

</td>
</tr>
</table>

The LEGO Ideas Book by Daniel Lipkowitz • Tips, tricks, and ideas for Lego creations. • $23.39	Alphabet Book of Lego Creations by Sean Kenney • An alphabet book with a different Lego creation for each letter. • $8.90	Premium Rainbow Stackable Base Plates - 12 Pack 10" x 10" Baseplate Bundle • Use these base plates to create a Lego wall or table, or let students use as portable working areas. • $59.99	Block Tech Sets • Cheaper but compatible with the name brand Lego; available in stores or online Walmart.ca; also available in single colour boxes, wheels, windows & doors, flowers & plants. • $9.97 each
 [1]Tutorial: Make your own Lego table using an IKEA Lack side table.	 [2]Tutorial: Create a Lego table and storage system using an IKEA Trofast shelf.	 [3]Tutorial: How to Make an Epic Lego Wall.	 [4]Free Download: Lego task cards to give students challenges of things to build.
 [5]Lego Education • Kits for using Lego in the classroom. • StoryStarters, Machines and Mechanisms, Math, Robotics, etc.	 [6]Numino Loops • Reusable, building block tape. • Works with brand name blocks. • Turn any surface into a building block base. • Comes in a variety of colours and two sizes. • Indiegogo Campaign that has	 [7]Free Downloads: Lowercase and Uppercase Alphabet Lego Cards.	 [8]Building Challenge

	raised more than 1.6 million dollars.		STEM and Writing Centers ● Challenge cards and writing center activities. ● $3.50

[1]Image from Lean and Green Frugal Living
[2]Image from That Crafty Jus
[3]Image from Renovated Learning
[4]Image from The Stem Laboratory
[5]Image from Lego Education
[6]Image from Indiegogo
[7]Image from Wildflower Ramblings
[8]Image from TeachersPayTeachers

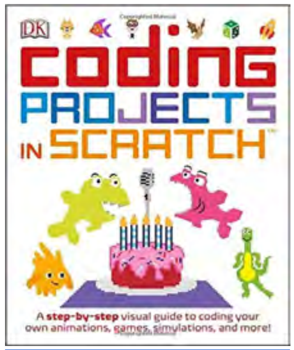
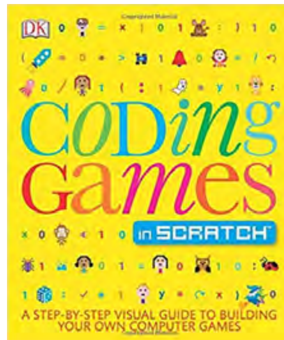
Robotics and Coding Resources

Robots and coding activities can range in prices, so start small and find something that works for your classroom, school, and budget. These are a variety of resources that can be used in early years classrooms and MakerSpaces.

 Learning Resources Stem Robot Mouse Coding Activity Set ● Make a maze and code the mouse to get the cheese by pressing the buttons. ● Ages 5-10	 Osmo Coding ● Block based coding game. ● Need an iPad and Osmo Game System. ● Ages 5-12	 Osmo Coding: Make Music and Jam ● Block based coding game. ● Combines coding and music making. ● Need an iPad and Osmo Game	 Modular Robotics Cubelets Six Robotic Kit ● A modular robot kit. ● Compatible with Legos. ● Grades PreK-12 ● $189.95

• $95.99	• $69.95	System. • Ages 5-12 • $79.99	
 Sphero SPRK+ STEAM Educational Robot • Block based coding to program the robot to do different tricks. • Grades K-12 • $164.99	 Toysmith 4M Doodling Robot • Build a vibrating and spinning robot that draws. • Ages 9+ • $28.93	 [1]Bee-Bot Programmable Floor Robot • Program the robot to move in sequences • Different floor mats can be purchased to work with the robot. • Ages 3+ • $123	 Ozobot • Robot that reads colour codes. • Grades K-12 • $89.99
 Wonder Workshop Dash Robot • Use the app to learn block based code to program the robot. • Ages 6+ • $229.95	 Coji The Coding Robot • Program the robot on the app using the language of emoji's. • Ages 4-7 • $49.99	 Bloxels • Learn how to create video games. • Use the physical board and blocks to create characters or rooms for your game. • Ages 8+ • $44.20	 Makeblock DIY mBot • Make your own robot and code it with Makeblock software, based on Scratch • Electronic parts are based on Arduino • Ages 8+ • $109.99

 [2]Cubetto ● On Kickstarter right now (released Sept 2017) ●Learn basic coding through hands on blocks ●Montessori approved, LOGO Turtle inspired ● Ages 3+ ● $195+ US	 Hexbug Nano ● A micro robot that propels itself through vibrations; can create mazes for it ● Ages 3+ ● $8.32	 Edison: Educational Robot V2.0 ● The robot can scan barcodes or download software to learn new skills ● Compatible with Lego ● Ages 5+ ● $49.99	[3]Hour of Code Website: https://code.org/learn ● Different levels and different topics for every age, ability, and interest of students. ● Great way to introduce coding.
 Scratch Jr. App ● Block based coding game. ● Ages 6-8 ● Free	 Bee-Bot App ● Based on the Bee-Bot floor robot ● Program the bee to move in sequences. ● Ages 4+ ● Free	 Lightbot Jr: Coding Puzzles App ● Simple symbol based coding puzzles. ● Ages 4-8 ● $3.99	 Lightbot: Programming Puzzles App ● More complex symbol based coding puzzles. ● Ages 9-11 ● $3.99
 GoldieBlox: Adventures in Coding App	 The Foos Coding App ● Symbol based	 Kodable App	 Nancy Drew Codes and Clues Mystery App

● Symbol based coding game. ● Ages 6-8 ● $3.99	coding game. ● Ages 5+ ● Free	● Start with an easy symbol based coding and move into JavaScript. ● Ages 4-11 ● Free	● Story based, puzzle game where kids have to learn basic code. ● Ages 6-8 ● Free
 How to Code: A Step-By-Step Guide to Computer Coding by Max Wainewright ● A step-by-step guide to learn basic coding skills and concepts. ● $15.71	 Coding Projects in Scratch by Jon Woodcock ● A visual guide for Scratch coding projects. ● $23.93	 Coding Games in Scratch by Jon Woodcock ● A visual guide for coding games in Scratch. ● $20.91	 Scratch Coding Cards: Creative Coding Activities for Kids by Natalie Rusk ● 75 activity cards to create a variety of programming projects. ● $26.94
 DK Readers L2: Story of Coding by James Floyd Kelly ● Easy reader that teaches about the history of	 Hello Ruby: Adventures in Coding by Linda Liukas ● Introduction to	 Robot Wars Coding Board Game ● Interactive way to introduce coding to kids. Commands for	 Robot Turtles Board Game ● Teaches pre-programming skills. ● Inspired by the Logo language. ● Ages 4+ ● $36.99

computers and coding. ● $4.94	programming concepts. ● $19.31	bots are written in Java. ● Ages 7+ ● $26.99	

[1]Image from Canadian Classroom
[2]Image from Kickstarter.com
[2]Image from Hour of Code
App images from iTunes.com

Building Toys

MakerSpaces don't have to be all about creating a physical product that they can keep forever. Students can make non-permanent creations using a variety of building and construction toys, such as these. Just don't forget to have them take a picture of their creation and save it to their digital portfolio first before cleaning up!

 Keva Structures ● Simple wooden planks for building structures. ● 200 planks ● $60.40	 K'nex ● Rods, connectors, wheels, etc. ● 375 pieces ● $26.59	 Lincoln Logs ● Wooden building logs for stacking and creating. ● 137 pieces ● $53.00	 Bristle Blocks ● 85 pieces ● $40.24
	 Tinker Toy ● Spools, rods,	 Play-Doh Variety Pack ● 20 cans	 Smart Dough Tools

Gears Lights & Action • Build moving machines with these gears. • 121 pieces • $69.76	flags, washers, end caps, and more for building. • 150 pieces • $65.00	• $24.95	Kit • 26 pieces • $15.99
 Brain Flakes • 360 pieces • $20.99	 Playstix • 150 pieces • $31.51	 Magnetic Building Toys • 64 pieces • $36.99	 Straws & Connectors • 705 pieces • $59.99
 Wooden Blocks • 100 pieces • $19.99	 Snap Circuits Jr. • 30+ pieces • $41.21	 [1]Squishy Circuits Standard Kit • Create electrical circuits using conductive dough, batteries, LED lights, motors, etc. • Standard Kit: $30	 Lite Brite Magic Screen • 200 pieces • $49.80

 Locktagons ● 200 pieces ● $34.80	 GoldieBlox and The Builder's Survival Kit ● Building toys designed to target girls to encourage them to go into engineering. ● $42.26 ● Lots of other GoldieBlox sets can be found here.	[2]PBS Kids Engineering Games Website: http://pbskids.org/games/engineering/ ● Lots of different engineering games, such as Inventor's Workshop, Jet's Rocket Ship Creator, and more!	 Smart Stick Plastic Blocks ● 240 pieces ● $12.99
 ThinkFun Maker Studio Gears Building Kit ● A little building kit to practice using gears, such as a car or motorcycle ● Ages 7+ ● $24.99	 ThinkFun Maker Studio Winches Building Kit ● A little building kit to practice building something with a winch ● Ages 7+ ● $24.99	 ThinkFun Maker Studio Propellers Building Kit ● A little building kit to practice building something with propellers ● Ages 7+ ● $24.99	 Thames & Kosmos Kids First Intro to Engineering Kit ● A beginner's engineering kit with experiment and building challenge ideas ● See more STEM kits by the company here ● Ages 5+ ● $49.95

[1]*Image from Squishy Circuits*
[2]*Image from PBS Kids*

iPad Apps for Students

Here are some iPad apps that you can use in your classroom or MakerSpace. There are tools to use in the actual designing or games that work on problem solving and engineering skills.

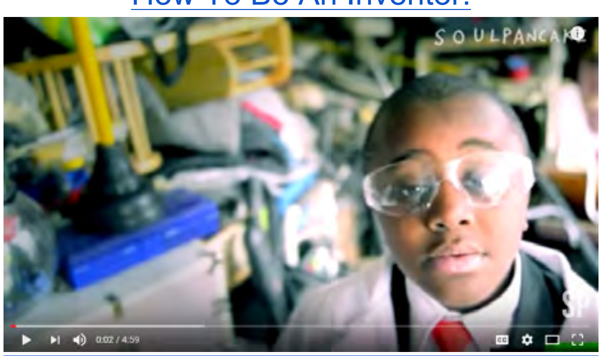
*Draw and Tell HD ● Students can draw their design plans and record their voice describing their plans. ● Free	*Pic Collage Kids ● Create design plans with the drawing tools. ● Free	Shadow Puppet Edu ● Take a picture and record your voice on top. ● Free	Seesaw ● Digital online portfolio. Students can upload photos or videos of their projects. ● Free
Simple Physics ● Design and build strctures, such as tree houses and ferris wheels ● Ages 9+ ● $2.79	Inventioneers Full Version ● Create your own fun, crazy inventions with the help of the inventioneers. ● Ages 6-8 ● $3.99	Crazy Gears ● Puzzle game featuring gears. ● Ages 6-8 ● $5.49	The Everythng Machine ● Design and make machines. ● Ages 9-11 ● $3.99 ● Many other similiar making or exploration apps here
Odd Bot Out	Green Screen	GoldieBlox and the	Sock Puppets

• Help the robot escape by using building blocks, electricity, and physics. • $5.49	• A simple to use green-screen movie making app. Great for student projects • $3.99	Movie Machine • Easy to create digitally animated short movies • Free	• Create lyp-synched animated movies • Free

App images from iTunes.com

YouTube Videos

Here are some good YouTube videos and channels to show your students for some inspiration!

Caine's Arcade Everyone needs to watch this inspiring video of how Caine created a whole arcade 	A Pep Talk from Kid President to You 
How To Be An Inventor! 	All About Cool Inventions! | Nat Geo Kids Cool Inventions Playlist 
Joseph Herscher on Sesame Street	John Spencer's Video Playlist

Kinetic artist Joseph Herscher sets up a Rube Goldberg machine to water a plant. An interesting video to introduce young students to the idea of Rube Goldberg machines. 	Co-author of the book “Launch” started creating video prompts for creative writing and maker challenges. 
"Code Stars" - Short Film A good introduction into what coding is and why it’s important. Check out the code.org channel for more awesome videos and tutorials! 	Hour of Code Message From the International Space Station NASA Astronauts Mike Hopkins and Rick Mastraccio explain how coding is important to their jobs! 
Above and Beyond A great introduction into the four C’s for students: creativity, communication, collaboration, and critical thinking. (Thanks to Shauna Cornwell for showing this to us!)	Kunal Nayyar Helps Grover fix his Robot A cute introduction video into robots and how they need to be programmed to do things.

How High Can You Stack?

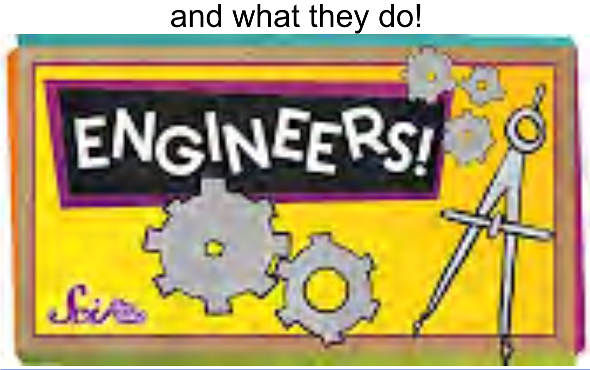
Students learn how engineering can help them build things better and stronger.

Think Like an Engineer Part 1

Learn how to think about and solve problems like an engineer.

Solve Problems: Be An Engineer

Learn about what engineers are and what they do!



What's An Engineer?

Learn about what engineers do and why they do it.

SciShow Kids YouTube Channel

Find a ton more science and engineering videos here on their channel.

Crash Course Kids YouTube Channel

Lots more engineering and science related videos on their channel.

BrainPop Jr. Requires a subscription, but has awesome science videos and games on it.	Design Squad Global Lots of videos about designing, building and creating inventions, and some challenge ideas!

Chapter 6 - Adjusting Delivery To Infuse Student Engagement and Purpose

By Peggy Hobson

Five summers ago, in August, a grade 6 student emailed me at home asking if we could purchase a 3D printer for the school. We emailed back and forth a few times and agreed to meet on the first day of school at lunch. He mentioned a few other students were interested in joining us for the meeting. On that first day of school, eight students, ranging from grades 5 to 8, met me in the media center and enthusiastically shared all the reasons why it was time for a 3D printer at the school. The student who sent the original email said, “Do you know, Ms. Hobson, there is now a 3D printer designed to print pieces of itself you can put together to make another 3D printer. In fact, it can print itself to infinity.”

I told the group my mind just snapped a little bit with that news, and yes, it was definitely time for us to explore 3D printers. I provided the students a budget of $1500 and they went to work, quickly and with passion and purpose. With all the due diligence one would hope for, these students researched for a few months through the world of 3D printers and came up with a report for me including their recommendation for the best model for the school, under budget. We purchased a 3D printer in November that year. And actually even without the purchase, the student learning could have been documented. When it arrived, the 3D printer needed to be assembled.

The “original 8” assembled that 3D printer and had it working within a few days. Their engagement continued along with their growing knowledge and application skills. Collaboratively the students learned about calibration and many other details. In fact, they even developed an equation for “actual real print time” noting the print time predicted by the printer instructions, was inaccurate. The equation (and a few working drafts) were visible on the whiteboard in the same room as the 3D printer.

These students’ demonstrated working knowledge and criteria for performance for a 3D printer alongside were outside any curricular outcomes they were exposed to in their classes. This was because they wanted to do this. A few teachers had expressed interest and had visited the room to see the 3D printer, but had not really “rolled up their sleeves”. The students presented, with pride, the 3D printer and the objects they had designed so far, and successfully printed, explaining their failed attempts and second or third trials, (photo above is of the first printed figure at the school) at a staff meeting. The staff listened and observed with keen interest, hearing about and seeing initiatives they had never experienced in their lives.

4D printing advancing globally

I am currently reading a book “The Great Disruption” by Rick Smith and Mitch Free. Smith and Free describe the current use of 3D print technology, the second wave of the industrial revolution and the disruptive innovations impacting global manufacturing and delivery. As I align my thoughts for developing this chapter so many of their points resonate with me and my educational perspective.

The authors provide numerous examples where, globally, individuals now have access to design software and the ability to print industrial parts without prerequisite course work, training or the experience we have come to expect in current education and training institutions. These individuals have a passion or identify a need. People are currently creating, designing, developing and printing significant parts at home. Smith and Free discuss 3D printing leveling the playing field and in some cases advancing new methods ahead of traditional practice in time, structure and costs. Conventional manufacturing is being challenged in our on-demand society. The potential for 3D printing is explained, as well as its significant impact on manufacturing and delivery.

Smith and Free take us into the world of possibilities with the onset of 4D printing where we can only imagine printed objects capable of rearranging themselves. As an example - 4D printed rubber with the internal capability to change tire density based on road conditions. There are new ways of doing things all around us. This change process leads to concern, discomfort and confusion for people stuck in their typical educated and experienced ways of accomplishing tasks. Things have worked for them until “now”. For the innovators, these have been and will continue to be exciting times where people are successful in design and creation of things, apps or new ways of doing things, because they can. This is impacting every sector we know.

Connecting school with authentic learning

Children in the world today face challenges, some of them life threatening. Many of these challenges are signs of the current global realities. They are growing up in a childhood world without the same restrictions we had. Restrictions are placed on them by the group norms where they live. They learn those and they become embedded. For a while, though, some only see possibility. Resilience. And for increasing numbers, as children see past the limits, they create something different because they want to and or able to; they see a solution to an issue they would like to impact. We need to encourage them.

The makerspace area in our school began to develop, facilitated by Brandi Bartok, our Teacher Librarian. Students continued to use the 3D printers (there are now two), the Arduinos https://www.arduino.cc/, Spheros http://www.sphero.com/ and the Raspberry Pi's https://www.raspberrypi.org/. A knitting club grew in popularity and there was a huge response to the contest "create something you can wear". Rolls of duct tape of many colours were visible on the shelves in that area. Teachers were beginning to see the benefits of exploration and making, but clearly this makerspace remained outside the perspective of necessary program delivery. It was engaging, drew students in, but was definitely "extra."

The program "Scratch" https://youtu.be/0pxaFzRtx7k, the Ozobots https://youtu.be/m5d4iXGbIGs, sites like "Learn to Code" https://youtu.be/fwIrAzZfvRc and exemplars (Middle Grade Genius Hour Project https://youtu.be/536MsLhLpe4) were available along with many more resources. Why not?

Noting the 3D printer was being accessed by a limited group of students, a 3D printing activity, a challenge, was initiated by a group of grade 7 students at the school. The challenge advertised was to design a model of a building on a 10X10 cm square using the app Tinkercad https://www.tinkercad.com/ and then have it printed. The collection of printed models formed a model town. The students set up a tutorial and links on a website that could be accessed through the school media center website. Interested students accessed the site anywhere, anytime, designed their building and submitted the file through email. The designs sent in were placed in a queue for printing. As the "town" grew, new issues arose that had not been considered. Learning. For example, the issue of scale was visible. The town's donut store was much larger than the multi-story office building. This led to deep conversations amongst the students creating the need for detail change in the project. The development of this town threaded through a school year. Although teachers visiting the media center could see the growing collection of buildings and seemed pleased with this idea, not one teacher designed or sent a building to the print queue. Not one teacher acknowledged the obvious curricular outcomes the students were achieving. The entire project was noteworthy but outside their program, seen as extracurricular.

A grade 5 student used his skills in coding (skills he developed due to his motivation, interest and engagement in seeking online tutorials) to write a script to develop a video clip to demonstrating the development of a black hole. The assignment in his class was to do a project on one section of the Solar System unit the class had been studying. His presentation and his

visual, the video demonstration of the black hole developing, was thorough, interesting and accurate. His mature explanation still resonated with me today. “It doesn’t matter if my teacher understands coding. What matters is that the information I presented is correct. I know I did not have to spend the hours I did creating the video. That was my choice. Have I demonstrated understanding of the outcomes my teacher identified for this unit? That is what will concern my teacher.”

Another student, inspired by the Raspberry Pis, became an expert. He programmed one to blink a small light in a circuit every time the International Space Station was over Winnipeg. He would be in class, working on an assignment, and he would simply put his hand up and state “space station overhead” and then go back to his work.

The above are examples of projects occurring at the school with engaged students who are involved in the process of learning many things. Not one those things was connected, by teachers, to achieving outcomes or assessment. The students were stepping up and creating and learning because they wanted to be involved.. It was not until two grade 7 girls, during their novel study, printed what they imagined to be replica model of the house described in the book they were reading. This was a connection to a class assignment bringing to life the idea that teachers would choose grade level content and students could choose the way in which they wanted to demonstrate their learning. But there is still a disconnect. The teacher checked the accuracy of the model house with the novel text but did not consider the design or production process or the outcomes the girls achieved achieved in a number of other subject areas. Two grade 7 girls had independently designed and printed a replica of a house!

Student Technology Leadership

A Student Technology Leadership Team formed at the middle school. Students stepped up and participated as they wished. One opportunity for the group was to keynote at a major professional development session in the city. The students prepared. Actually, as a collective, they came into the school during one of their days off, a school administrative day, requesting a room in which to work. The HGI Student Technology Leaders collaboratively prepared their presentation, which did include multimedia: https://youtu.be/212lvShWvus

The group was mature and simply awesome during the professional development session. The team of students cooperatively and at a high level, presented relevant information, challenged the adults and engaged them in learning. They led an inspiring questions and answer session. In the exit slips, a few participants commented the session would have been better if the students had not been handed a script to work from. The students, although concerned about the feedback because it was so inaccurate, did take it as a compliment, a commentary on how good they must have been. Once again, I was concerned about the boxes in which my colleagues lived.

Outcome based instruction

My focus, connected with purpose, shifted again. And again. How could we, as educators of these students, assess the outcomes they had so obviously achieve, even though the problem or some of the solution strategies are not delivered to them by us nor was their work along the way observed by us? If students are invited, encouraged and "let go" to explore their world and their passions, would they cover necessary concepts to become literate and numerate?

Dialogue regarding educational change surrounds us. The difficulty is that we are embedded in our perspective of education and we have difficulty stepping out of it. We hear about innovations, we attend professional development sessions and we are intrigued with the information. We are motivated, care deeply for our students and their success. We are eager to deliver the best education. We are "stuck" in a system of little boxes (timetables, rows of desks, attendance sheets, grade books and classrooms) that were embedded in traditional for reasons responding to the societal needs and structure of the time. We know that system. We attended school in that system. The words "think outside the box" are indeed motivating to us and we understand on some levels what that means. We have even experienced the stepping out. But while we are mired in a complex system of overlapping boxes, we have difficulty really moving away from the box, or more importantly, the series of boxes. Examples of educational change are visible and implanted in many, many places and these are growing in isolated pockets or growing collaboratives. Working models, engaging programs, shifts from traditional delivery are present, available to view on social media, in educational journals or down the street. Creative teams have embedded their philosophy into the development of schools or program actualizing the tenants of inquiry, student voice, and possibilities. The critical reasons for adjusting program delivery are surfacing globally as there is a realization our population cannot rely on the responses or the actions to which we have grown accustomed. Habits. Our habits or our boxes are not leading to real solutions or a better way. With the opportunity to watch and listen to our students, we can see another direction if we guide them and follow their lead.

Change; moving away from home

Recently, an unexpected opportunity came up for me to travel to visit my sister at her summer retreat in the Adirondacks in New York State. My brother was also planning to arrive. The three of us had not been together in years. After typical summer days at home, I was pleased to have this adventure fall into place.

As I packed and prepared the day prior, I thought, “what if I just stay home?” It would be easier.

Although it was exciting to think about the travel, the reconnecting, the laughter and memories, home was so very familiar and predictable. This was not even an example of a big risk.

I boarded the plane. Hours later I was together with my siblings on the shores of Long Lake. It was simply beautiful. My feelings confirmed, at that moment, was that this was the right thing, even though my more predictable days gave way now to the unexpected. Each day of the visit was filled with sharing stories and perspective, stunning scenery, something new to try and to learn. There was comfort and there was adventure.

Although this decision to travel, enjoy, learn and breathe in the differences of another community was on a small scale, the tension in this decision was so similar to what happens on a larger scale at school. As the teaching team prepares with excitement for the school year, the planning must include intentional steps to ensure there is a move away from “repackaging lessons we have done before”. This can be compared to “what if I just stay home?” Moving toward intentional learning and change. Indeed it is always easier to do what we know. Students would agree as well. Yet students are willing participants in the activities they choose to do or to which they are drawn. They are at the age of experience. They do not have as many preconceptions. They are up for the adventure. And many are easily led into the “worksheet mentality” of just waiting for someone to tell them what to do.

Authentic experience and inquiry

There is certainly a prevalent dialogue about the differences in teacher directed lessons and student centered or student inquiry. Although the positives and the successes for the latter are well documented, a number of teaching teams sometimes still “stay at home” when planning. The expectations of what we “know to do as teachers” are so well engrained. To stay at home, though, does not allow one to encounter the results in those motivating different experiences. And those experiences are quickly being recognized as what is globally necessary. Disconnect.

Grade 6 students used "MineCraft" https://minecraft.net/en-us/ to create replicas of World War 1 trenches for their inquiry project. After they completed their work, they interviewed a veteran able to provide insight and who corrected the boys' conceptual understanding. Driving back from the interview, the boys accurately corrected the trenches, on their devices, in the car.

An annual grade 8 project is to create a model of a cell as a conclusion to the unit. In the last few years, this activity has changed from all students asked to use candy to create the cell to developing a model with choice of materials. It was not long before a student learned how to operate the 3D printer to create her model.

To draw students in, to follow student's lead, to unleash the hatching interests, to embed risk, trials and success in whatever problem students are seeking to solve or whatever creative endeavor they are savoring becomes the engaging way to facilitate. The challenge to switch our focus away from "home" becomes the observation of student inquiry and the path of discovery. To do this, the ability to affirm the curricular outcomes in which the student exhibits understanding is part of the journey. We are not "home", holding on to the outcomes we know and have decided to deliver. We are venturing to observe an explorer and together facilitate this personal path of meaning. Learning.

We, as educators, can articulate the move (and explain why the move is necessary) away from the "sage on the stage" and we can define that very method of delivery as traditional. Yet, we are caught. We are inside Ruben Puentedura's "SAMR model"
ourselves, struggling as a group to step intentionally into the two levels, modification and redefinition, at the right side of the continuum. All four levels are critical and intertwined.
https://www.youtube.com/watch?v=9b5yvgKQdqE
https://www.youtube.com/watch?v=ZQTx2UQQvbU
http://www.hippasus.com/rrpweblog/archives/2014/06/29/LearningTechnologySAMRModel.pdf

Coding

Maybe it was not until Hadi Partovi, through code.org, initiated "Hour of Code" that a number educators sat up and really noticed coding. In well thought out advertising and distribution, the number of participants in the Hour of Code increased each year, globally. Coding courses and apps began to populate student devices. A student keenly interested in this area could learn independently. Options increased. Some schools dabbled in the "hour" each year and did not take this idea much further. Some schools began to infuse coding into their programs. Even though there was not a curriculum document teachers began to connect the activities to outcomes. Some students began drawing an income from their independent work or from monetizing apps. Scott McLeod says it best in one of my favorite TedTalks: https://youtu.be/GyIl4y_MRbU

There is a continuum of skills in coding, consistent with most learning. The Hour of Code presented engaging activities for all levels of learner. Because this activity was unevenly distributed in its infancy, being "found" by educators more connected on social media, it was seen as an "extra". Many did not see the embedded skills or recognize it as anything more than an interesting activity. An hour covered it. As the following grew each year and the publicity was more widely distributed, both students and teachers began to look forward to the hour. The implementation of this hour of code aligned with the growth of the maker movement. Most students benefitted from these innovations as they were engaged, brought passion and ideas to both. Inquiry based delivery was also developing in pockets in increasing numbers of educational settings. The idea of "extra" remained as teachers were still covering the traditional program curriculum and hesitant to recognize or look for links where these innovative details meshed with program outcomes. These teachers preferred to "stay home".

Today there is a definite continuum of delivery strategies and philosophies in our schools. There is a range from dabbling in new strategies to complete embedding of new ways to do school. In a world of disruptive innovation, changing industry standards, self-driving cars, medical interventions and documents like the Conference Board of Canada's innovation skills http://www.conferenceboard.ca/cbi/innovationskills.aspx we decided to move away from dabbling and immerse ourselves.

In a focused attempt to move away from some boxes, our middle school has successfully developed and implemented FLEX time. Each grade level team has a half day where there are no specialist periods scheduled. If there are four classes, four teachers, there are corresponding choices. Basic level – teachers are able to work with their students all afternoon on activities in the order students choose. Students know they have the paragraph to complete, the mathematics quiz for which to prepare and that social studies project. Students pick the order they will complete their tasks and the people they will need to see for assistance if needed. Moving up the continuum – at their grade level team meeting teachers prepared workshops based on the curricular outcomes presenting difficulty for some students. They compare notes and offer workshops during FLEX for any student in the grade who needs to or want to attend.

Students, now not limited by boxes, may make an informed decision about which workshop to attend. If possible, another teacher (examples: Teacher Librarian; Applied Arts Teacher) or two are aligned during the half day to increase the facilitation. It is a bonus if one of the school's applied arts rooms is available. Highest level – the entire grade group works on passion or inquiry projects during the half days designated as FLEX. Students, individually or in groups, identity an interest or a problem that needs to be solved. They dialogue, plan, research, contact experts, visit someone in the field, curate, create, present and learn by a process that is facilitated by the teacher team through mini conferences, questioning and guidance or simply knowing when to let go. We intentionally attempt to remove limits. AI, VR, coding, making, exploration at any level is encouraged. The process of learning is at the center.

FLEX is now not an extra at the school. It is embedded time for learning. The next step is for the teaching team to enhance is the recognition of which outcomes are accomplished by which students during this messy and wonderful time. These are exciting times to create an environment where everyone is accepted, every contribution is welcomed and we navigate together to create citizens the world will require with the necessary skills and confidence to help define their success and articulate how this will be a positive, meaningful step or process.

Another initiative at the middle school is Interactive Start. After "O Canada" and attendance each morning, the entire population of the school engages in activities and/or passions for 30 minutes. Although there were bumps in the implementation (example: establishing this time as a time for true choice, not using this as extra time for team practices or choir rehearsals as then there would be an expectation for attendance) we have a joy filled time each day where people engage with each other in common interests. Participants can participate in the same activity for a number of days, or try something new each day.

Interactive Start has developed far beyond our expectations. The benefits are many. In a community where a large percentage of students are driven to school and directly from school to an extracurricular event, there is not the time in the day to chat with their peers except during the "fast" locker breaks or lunch time. This time in the morning is seen as an invitational one with many options that do not come with pressure. Adults and students engage together. Students facilitate sessions. Adults participate. Adults facilitate activities. Students participate. Or together people participate in an active game or physical activity like a run around the community. The keys in this tension-free time are many. Participation, relationship building, learning about individual differences and healthy interactions while pursuing passions. Although many curricular outcomes are covered during this time, to track them would result in the negative of trying catch something that is free flowing.

One of the passions during these morning session is coding. Coding for coding sake, a coding course to achieve an online certificate, or coding with a purpose connected to a project. People who share this interest work together to build their skills or work on a personal initiative in a parallel way. They ask for assistance. They share ideas collaboratively. They are becoming the citizens the world will need. The participants are reading, sometimes at high levels. We focus on literacy levels sometimes in prescriptive ways, yet we have seen a child, identified as a struggling reader, pour over text to learn the instructions for his next step in a coding program. Meaning.

Coding intermingles with so many curricular outcomes and multi-modal literacies. On the horizons of the more uncertain educational delivery we need to encourage students to follow their passions and their questions and their creativity. We need to let the learning environment get messy and figure out strong ways to encourage and enhance. We need to be bold and accept that we do not hold all the information and that often a student may create a "black hole video". And if the creation or the presentation comes back as inaccurate, we need to practice our best facilitation skills. To not do this is to squash the important learning process.

As our school is a device friendly environment, students can use technology to curate or to create. As the understanding that teachers will focus on identified outcomes and encourage students to decide how to demonstrate their learning becomes entrenched, coding is increasingly chosen as a strategy for student presentation at the culmination of a project. Sir Ken Robinson's words still ring with significance. We as educators must make sure we are doing more than watching this TedTalk: https://www.ted.com/talks/ken_robinson_says_schools_kill_creativity Looking at the realities and predictions of employability skills and innovative skills, to not notice the critical need for people who have significant knowledge in coding does a disservice to our educational model. Infusing authentic problems (products, services, strategies) into our schools will move students into meaningful learning and creating.

The more choice students have, the more they will work, learn and teach. The more they will connect their learning to authentic work. They will develop positive self-concepts, problem solving skills, collaborative skills and positive relationships with a diverse group of peers. Their voices will be significant and they will feel confident sharing. They will understand that a mistake or a failed attempt will be part of the creative process and they will continue toward their goal. In this, they will feel accepted, capable and understood. These will be the people who become the very citizens we need and look to for guidance in these complex global times.

We are dabbling in adjustments and changes to our educational models. Yet these models are embedded in so many agendas, we have lost our collective way and do not have a clear path to the outcomes we would articulate together. We must collaboratively take that step to create inviting and accepting learning environments intentionally for the citizens of tomorrow – we are going to need them. We must, in many functional and exciting ways, provide an environment where our students collaborate with us to become their best selves as citizens ready for the global realities of a 4D world and beyond. And the students are waiting for us to figure this out or are disengaging from our current educational system in frustration.

Let’s not just “stay home” anymore.

Peggy Hobson
Parent, Educator, Gardener, Photographer

Peggy is the principal of Henry G. Izatt Middle School in the Pembina Trails School Division. This middle school has a BYOD environment. She facilitates innovative learning practices embedding inquiry, technology, digital literacy, global citizenship and creativity through infusion of student voice. She is a recipient of The Learning Partnership’s Canada’s Outstanding Principals award.
She has facilitated the organization of EdCampWpg, the Pembina Trails Student Code Camp and has been a member of ManACE.
Along with the Student Technology Leaders from Henry G. Izatt, she looks for opportunities to share ideas with others to empower teams to facilitate creative learning environments.

Linkedin: https://ca.linkedin.com/in/peggy-hobson-17a0b638
Twitter: https://twitter.com/peghob
Website: http://peggyahobson.weebly.com/
Blog: http://phobson.wordpress.com/

Chapter 7 - Creating a Makerspace - A Functional Plan for Maker-Philosophy for Grade 7

By Chad Wilson

Philosophy and Rationale

When we talk about makerspaces we generally refer to the physical space/s or classes that house the tools students will work with. We have seen multiple classrooms in different schools that are physically set up as "makerspaces", or places where teachers will bring their students to actively engage in making. The purpose of these classes are to explore different ideas and tools and eventually build/create, occasionally with a direct product in mind but often creating/building out of exploration.

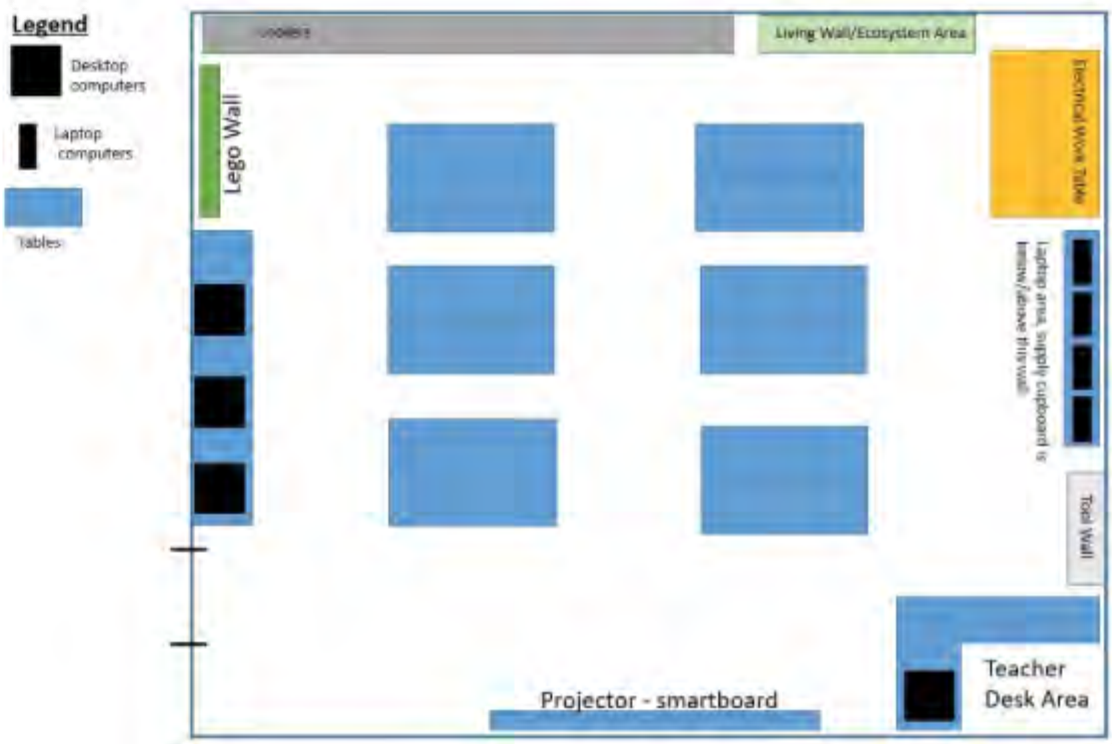
As teachers we can take the makerspace idea further. These experiences shouldn't be limited to trips to the library or design build challenges, they should be a routine part of school, and used to help students connect with the curriculum we are trying to cover and the skills we are trying to help them develop. Jean Piaget's theory of Constructivism describes that learners construct their own knowledge through experiences. They learn new information not by being told it, but by experiencing it and making connections. In middle years environments we often find ourselves teaching our students multiple classes, or sharing the responsibility with switch teachers. This provides us with the time and freedom we need to bring the makerspace idea out of the library and make it an active component of our classrooms.

The following are plans to amalgamate previous spaces and tools in Leila North School towards creating a makerspace for my own personal classroom, and the school at large. This project will also analyze the curriculum to determine how a makerspace philosophy can be seamlessly integrated into the regular day and programming of the students. Since I will be teaching in a grade seven classroom at Leila North next year, these plans will be put into action in various degrees next year. I have always taught in a cross-curricular, project focussed manner, blending courses and concepts together to make a more holistic learning experience for my students. It has always been my belief that this would allow students to connect with material more easily, and it would also help them understand that everything was connected - in life there is no "science" that is separate from "social studies", or math that is separate from language arts. In daily life and in our jobs we have an understanding of how things work together, which is what I wanted to develop in my students. Taking classes discussing the philosophy of Piaget, Dewey, and Papert has furthered my understanding of why cross curricular, project based learning works with students. Their personal experiences are what causes the learning to take place. Without the experience, material is abstract and exists only in the classroom. I have decided to move back to grade 7 at Leila North next year to further explore the design of a complete educational experience for students designed around the philosophy and ideas listed above. My idea of education is one in which students wonder, dream, design, create ask questions, and develop skills in addition to those found in foundations for implementation documents.

In Seven Oaks we are lucky enough to have stand-alone classrooms at the middle years level where we teach one class all subjects. I say we are lucky because this allows for cross-curricular education, and more importantly the opportunity to make maker philosophy a reality on a daily basis. These plans encompass both the idea of a physical makerspace, and the idea of using maker/tinker/engineer/project learning ideas. They are designed with the student in mind - to create learning experiences with which they can anchor curricular concepts. This will create a deeper, more personal understanding of concepts for the students while also encouraging the development of other skills such as the 6 C's (Character, Citizenship, Communication, Critical Thinking, Collaboration, Creativity) - which are an increasingly important aspect of education within Seven Oaks.

Location

There are two aspects to this makerspace: using the classroom as a permanent makerspace, and using the school as a whole as a makerspace.



The Classroom as a Makerspace:

The above image is a diagram of the proposed classroom. The only additions to the room are the lego wall, the living wall/ecosystem area, and the electrical tables (these are repurposed lab tables that can double as experiment tables if need be). The classroom itself will function as a makerspace in that it will house the majority of the tools and supplies the students will be using. As students and I progress through learning experiences there will be a consistent need to explore, design, and create with the material we have available in the classroom. These materials will be stored in a corner of the classroom in labelled bins. I will be teaching in a classroom with 22 students, in a room set up in a table formation.

Classroom Features:

Tables	I have chosen a double table formation specifically to develop student communication/collaboration. As projects and designing will be a key part of the classroom, table groups are set up as work areas to facilitate communication and collaboration. For smaller groups the tables will be split up.
Desktops/laptops	The classroom will have the standard three classroom computers.It will also feature our repurposed laptops. This is to ensure that each group will have access to at least one computer if they so desire. Laptops will be able to be taken to tables.
Living wall/ecosystem	The purpose of the living wall/ecosystem will be for students to maintain plants and hopefully create and maintain a self-sustaining ecosystem. This will be a big part of our exploration of sustainable ecosystems and will be built/developed each year with a new classes.
Lego Wall	Lego walls are something that I have recently discovered. Although they do not serve as a direct link to concepts, they allow students to create/explore on their own and the lego can and will be used for prototyping. This is to encourage discovery, design, and creativity.
Tool Wall	The tool wall will house the specific tools for our makerspace. All dangerous tools (exacto-knives etc.) will be locked away in the desk drawer when not in use.
Electrical table	The electrical work table will house the majority of the tools required for the Arduino, as well as the soldering tool and cabling.
Portable White Boards	The room will also feature portable whiteboards. These will be able to be secured on walls or taken down to be used at tables. These are to help students with their designs in group formats and will serve as a key tool in project design/creation.

The School as a Makerspace:

Wood working shop	Leila North woodworking shop has a variety of table, skill, and band saws as well as lathe machines, a planer, drill presses, and sanders. This is used by the whole school for TAS (technology, arts, sports) classes however is not in use in the afternoons.
Community Garden	The community garden at Leila is a small growing area that is used for vegetables. Located in the back of the school it has the potential to be used throughout our learning experiences at the grade 7 level.
Graphics shop	The graphics shop at Leila includes screen printing machines and airbrushes. Currently there is no 3D printer in Leila North
Mac Lab/Computer lab	The Mac Lab contains 25 Apple desktop computers. These come complete with webcams and an audio station for recording. It is a bookable room. Leila North also contains a PC lab with 30 PC computers. This room is also bookable.
Bee hive facility	The beehives at Leila are currently located on the rooftop. They will play a part in ecosystem discovery, however they will also provide us with inspiration for project design and creation.

Resources and Budget

The budgeting for my makerspace will largely come from my classroom budget - set at a modest $250. In addition to the classroom budget we are also given a supply budget as each student pays the school $30 for supplies. With my supply budget I was able to purchase the glue guns/glue sticks, playdough, and LED's. With the classroom budget I will be able to buy the required parts for the lego wall, portable white boards, and the plants/structure for the living wall. I will also be relying on community donations/kijiji for other supplies, such as the lego itself and the K'nex.

I have already submitted a recommendation to the school to purchase the Arduino kits,and the extra laptops from Computers for Schools with the idea that I would become the school expert and reach out to other teachers to develop PD opportunities. If this request fails I will be looking to post ads on myclassneeds and gofundme to crowdfund the required money. My goal for the classroom will be to add 1 new piece of technology a year. This not only lets me stay within my budget, but it also allows me to explore the technology to feel comfortable enough using it in the classroom on a daily basis. This year's new technology for me will be Arduino, as I have no prior experience with it.

Physical Resources - Year 1

Resource	Received From	Price
Class resources	Glue, glue guns, cardboard, foamboard, tape, popsicle sticks, etc.	From art cupboard/supply budget
Various screwdrivers and screws	Community donated	Donation
4 Desktop PC's	Already installed in classroom	-
4 Laptops	From computers for schools	$400
Soldering Iron Kit x2	Amazon	$20 x 2
Soldering supplies	Donation from Maples Collegiate	Donated
LED's	Donated	School Science Supplies
Arduino	1 starter kit 3 regular MKR boards	$89 $34.99 x 3
Phone tripods	alibaba	$2/each
Webcams (4)	kijiji/community donated	<$50
Lego, K'nex	Donated/value village amazon	Donated, or $26.59 per box x 3
Playdough	Dollarama	$10
LED's	Dollarama, home depot	$20

Computer Programs

Computer programs will be loaded on pc's and laptops. All programs are free and will need to be installed by the tech department

Name	Description
Scratch	Used for basic beginning coding.
Minecraft	Used to practice coding/create worlds/games
Kodu	Basic Game design

Tinkercad	Our 3d modeling engine of choice
Blender	For enrichment - those students wanting to progress to more difficult modelling.
Sculptris	Subtractive 3d modelling for organic elements
Mblock	Used for converting scratch to code for Arduino
Stop motion studio	Used for our stop-motion activities
Arduino software	Used for programming the Arduino's

Community Communication

Club creation

Although the makerspace I create will be used by my classroom, I will be looking for students to participate in maker clubs at lunchtime. The lunchtime club will be centered around exploring some of the technology we have available to us (3D modelling, coding, Arduinos) with an aim of exploration. It will also provide me with further opportunity to explore the Arduino's - as they are my maker learning goal for the year. This club will also give me the opportunity to start a destination imagination team, however I will see if this fits into the program or if it is better left to an after school club. The rationale of the lunchtime program is not only to let students explore the technology, it is also to generate interest in the space as well. When students routinely use new technology in the classroom it does not go unnoticed by the administration or school community. This can lead not only to additional funding, but also spread awareness to other teachers in the building. In some ways it would suit my own goals but also increase the awareness for this type of education.

Community Outreach

Community communication can serve multiple purposes to my makerspace. Reaching out to local businesses and entrepreneur startups (Futurepreneur Manitoba, Northforge) can potentially result in fundraising, or even used equipment that is no longer needed. Reaching out to local businesses and professionals can also lead to learning partnerships where the professionals come and visit my class/clubs to discuss their use of modelling/coding/making. This not only generates interest in our current projects within my students, these people will also serve as role models showcasing potential careers in technologies that my students are interested in.

Community also means reaching out to other maker programs or similarly minded teachers. By joining forums such as the MAPLE forum or other online based maker forums/Google/Reddit/Twitter/Instagram groups, or attending workshops/pd based around technology my students and I will continue to find and share new ideas. An important part of this is also the maintaining of a classroom website/blog/instagram account to showcase what we are making etc. This is an excellent way to reach out to the community to build relationships and let others know where we are and what we are doing.

Schools themselves are communities, and more often than not great ideas and best teaching practices are not shared throughout the school. As a classroom makerspace it is imperative to let others in the building know what I am doing. This not only raises my profile (useful towards more funding from the division), but it also creates new ideas and raises awareness for this type of learning. At H.C. Avery I started as the only person doing stop motion routinely in my student projects. By the time I left I had my own TAS class based around graphics education, and the majority of the teachers had begun to use stop motion and 3D modelling in their class. This was the result of numerous formal and informal PD sessions put on by myself and others in the school who shared similar interests. The point being that if no one knows what you are doing, you have a harder time growing and developing a community not only of students, but other educators.

Classroom Integration

Explanation

The purpose of having my own makerspace in my classroom is to integrate the pedagogy of makerspaces directly into my daily teaching. By teaching in a manner which promotes design-based thinking and project-based learning, I will be in continuous use of my class as a makerspace. In addition to this I will also be encouraging the use of passion periods throughout my teaching weeks. These are periods designed to let students work on their own interest based projects. The overall aim is for my students to explore new ways of building and showcasing their learning while exploring new technologies (or at least those new to them), all in an effort to develop understanding of concepts we are covering and development of core skills, such as the 6 C's. Instituting this type of learning in the classroom must be done carefully as I have personally experienced the results of "I have a great idea, let's explore this…" in my early days of teaching. Although these moments of spontaneous teaching were created with the best of intentions, they did not set the students up for success. The following is how I would like to set my classroom up to make the most of our makerspace.

Stage 1: Team Building - understanding yourself as a learner

When students understand their own strengths and challenges, they begin to see themselves as contributors. Building confidence and educational identity in your students is a key part of a successful makerspace but it lets all students know they have something to add to group situations, or independent learning situations. This can help create the rise of sub-teachers in your room - students who know they have a skill in a certain area and see themselves as being able to help others they know have different skillsets. I like to start the year off by having my students explore multiple intelligences through various activities and even a “test” to help them realize where they fit on the multiple intelligences spectrum.

This will lead right into an activity about themselves. There are many ways to do this, but my intentions are to get the students making right away. The project will revolve around the idea that the students design something that showcases who they are: the influences in their life, who they are as a learner, their experiences, etc. Many students will choose to make a poster out of this, but others will turn to tech tools. Still others may consider art, drama, and yes….interpretive dance. The idea is to let the students explore what they can do in the class, the freedom they have, and the materials we have available to us while they create a simple project to tell others about themselves. This is essentially our first “make”. It will allow everyone to have some success, and they will learn valuable things about each other that will be helpful as we grow as a class together.

Stage 2: New tech introductions

When you are designing your class around maker/design-based thinking/project based learning, a key element is for students to learn new ways of building/making and expressing their understandings. Programs like Kodu, TinkerCAD, Arduinos all have varying degrees of difficulty levels and will require you to sit students down to teach them the basics of the program. This can be done in the lab, where the students can each have their own program running. I like to run through the very basics of a program before allowing the students to completely explore. A good strategy is to verbally encourage kids to check each other's ideas out by walking around the class. This encourages conversation, and stimulates creativity in the class as students see the ideas of others. Pulling up samples from around the room on the big screen and encouraging questions like “what would happen if you….” or “can you make it do this…..” also help foster further exploration and creativity. Eventually you will reach a time when the students who are drawn to the program have explored it enough to understand its potential and others who think it's neat but are not passionate about it are losing interest. This is completely alright, as the goal of new tech sessions like this is exposure and awareness. The goal is for students to have another “tool” with which they can create and make with. For my classroom I plan on introducing one new technology a week at the beginning of the year, beginning with simple programs like Kodu and working up to more complicated techs like Arduino. Every time a new tech is taught it will be done through guided understand and exploring/encouraging. Occasionally when the tech allows for it, exploration will come before guided instruction to allow for students to form a deeper personal understanding.

Stage 3: Good group work/collaboration

Although it may seem natural, many of our students at the grade 7 level have not yet developed positive group work skills. This needs to be addressed as group work/collaboration is a key element of my classroom. I love the idea of practicing VTS (visual thinking strategies) in the classroom, and I think it's perfect to reinforce what good collaboration looks like. This is also something that can happen during exploration activities, as it is key for students to understand what inquiry looks like!

The above ideas happen very early in the year. They are put in place to make my makerspace successful. Moving forward, I would like to institute maker ideas in multiple ways in my daily class life. The following are different ways these ideas will be developed in my class.

Classroom Implementation

Shoebox Challenge Fridays

I have long had the idea of presenting my students challenges that require them to work in teams to complete a task of some sort. The task would be open enough to allow maximum creativity but would have boundaries in place to maintain focus. I had no idea that this was not only a “thing”, but that it had its own global competition called Destination Imagination. In my version, students would receive a shoebox or something similar with supplies and a task. They would then have to work together to problem solve, design, make before showcasing to the rest of the class. This would be done on Friday afternoons, because let's face it - students want something fun, and they will look forward to it! The pedagogy behind this revolves around the idea of making for a purpose. As students work together in a situation like this they further develop their 6 C skills (especially communication, collaboration, critical thinking). Connecting this loosely to curriculum also encourages personal learning experiences to happen. Sample challenges include:

Design-Based Thinking Projects

Design-based thinking projects are an essential part of maker ideology because they teach empathy, and designing for a purpose. When teaching design-based thinking to students it is always easiest to jump in with partner discussions. This helps identify a need through discussions like “_____ would be easier if __________”. If partner discussion doesn’t work you could address topics like your school, local community, or city. This leads to the brainstorming phase where the problem has been identified and students seek out a solution. Finally, the team decides on a solution and begins to create it. Design-based thinking projects can be powerful because they connect the learner to a real problem that is occurring, and puts the ownership of helping solve the problem in the hands of the group. These projects can be quick, or they can be longer, more in depth problems that can span a unit or longer. The purpose of these projects in my class will be for students to join multiple curricula and skillsets together to tackle real-world issues and problems. They will be done in small groups, to larger groups. The emphasis is on empathy, understanding, design, and citizenship.

Maker Projects

Throughout the year we will be doing other maker projects in small groups and individually. These will be linked to the curriculum but with not have the emphasis on empathy that design based projects do. Their design is to create further experiences that link personal learning with the curriculum.

Sandbox Time

Sandbox time will be a fancy name for play time. Students will be encouraged to play with certain maker materials in our class - i.e. Lego, Arduino, Minecraft, etc. The goal of sandbox time is exploration, creation, and student choice. It is for students to further explore different options in the classroom, for them to explore their creativity in designing and building on a personal level, and finally, for them to stumbleupon ideas, interests, and passions. This is the precursor to student choice projects.

Student Choice Projects

Throughout the year students will get the opportunity to work on student choice projects. These are open-based projects designed to let students explore interests and passions. Play within the space of their interest will be heavily emphasized before students come up with a project they would like to complete. They will set the learning goals they would like to achieve, and design a project plan to follow. This will be reflected upon and changed as often as need be. The goal is for students to have a choice in their study, for them to take part in personally meaningful learning opportunities, and for them to understand project management and organization.

Assessment

Assessment will take both on a personal level by the students, and through discussion with myself. As students progress through combined units of study they will be progressing through a rubric complete with essential understandings and level of completion. These rubrics are progressive - meaning students gradually work their way up the scale. An example of a rubric for science topics in grade 4 that I created is listed below.

Subjects	Beginning (C-)	Approaching Expectations (C)	Fully Meeting Expectations (B)	Exceeding Expectations (A)
Science EU 1 - Light and sound are forms of energy in our life.	Student is able to identify forms of energy.	Student is able to describe why light and sound are forms of energy.	Student is able to demonstrate light and sound are forms of energy.	Student is able to analyze a discrepant event in terms of energy.
EU 2 - Light and sound can be described by properties and how they interact with materials in the environment.	Student is able to describe how light and sound interact with different materials and objects	Student can describe properties of light and sound as well as how each interacts with different materials.	Student is able to compare and contrast light and sound using their properties and how each interacts with different materials.	Student is able to create an experiment that demonstrates a materials ability to increase/decrease a characteristic of light or sound.
EU 3 - Light and sound can be used and manipulated to help us interact with our environment.	Student can give examples of how we use sound/light.	Student can describe how light/sound can be altered.	Student can describe how sound/light are being altered to help in a situation.	Student is able to create a device to solve a specific human need using sound/light.

I have chosen this style of rubric because when you would like to spend your time making/creating, you need to assess in ways that account for that time. Through recorded conversations, pictures, and video using the app Seesaw I will be able to create a portfolio for each student (eventually leading them to do it on their own) that showcases their progression through the scale while they take part in the making activities. The making process allows for conversations between teacher/student to occur that can yield far greater understanding of where the child is with their learning as students can describe their making objects and how they relate to the task at hand. Again, the personal connection between their idea and the concept being taught will be put on display. Assessment will ultimately be an ongoing conversation between myself as the teacher and the student with their creations as references for their understanding.

Units of Study and Maker Ideas

The following plans illustrate how I would like to split the year into 4 cross-curricular units. The topics have been chosen to be together in each case because of the similarities they share and the connections I will be making. They also exist because of the maker projects I will be doing to tie concepts together. Think of the maker projects as the glue that holds the different ideas together. The tech and maker tools taught to the students will help them explore, create, and make their projects. I fully expect students to design games on Kodu, build 3D models, etc. To get an understanding of how the project meets goals in each topic you may need to refer to the curricular documents. The plans do not represent all of the ideas or activities that will be done, just what I am currently working on.

Unit 1:

Science	Social	Math	ELA
The Earth’s Crust	Mapping the Globe	Number	Explorer writing

Rationale: This will serve as a good introductory unit for the year and will cover many essential understandings that will be needed moving forward. As students understand the earth's crust they will gain an understanding of why the earth is shaped the way it is. During this process basic concepts of number will be needed to numerically make sense of the earth. Finally, one creation project will see the students practice creative journal writing.

Project	Category	Description
Sensory Earth	Design-Based Thinking	Students experience the earth with their senses, most often visually. This design-based project will address the need for people with sensory impairment to see the world in their own way. Students will be required to create something that allows someone with a sight sensory impairment to understand the geography of the world (this could be climate, vegetation, topography, land masses, etc.)

Around the World in 80 days	Maker	Students will design vehicles with our maker equipment that will allow them to race around the world. They will be required to meet the challenge of collecting specific objects from different parts of the globe to ensure they don't choose the super rocket plan. For instance, students will need to step foot in the amazon to retrieve a certain plant etc.
Creative Journaling	Maker	Students will be required to make a creative journal of their travels around the world, with accuracy expected. This could be a podcast, handwritten, video, pictures, or something else.
Mineral extraction	Challenge	For a given new element, students will make a device to extract it from the ground. This will be a simulation activity. The mineral will be in a hard to reach location of the classroom.

Unit 2:

Science	Social	Math	ELA
Interactions within Ecosystems	Human Impact in the Americas	Patterns and relations	Analyzing news articles

Rationale: Human Impact in the Americas and Interactions Within Ecosystems go hand in hand. Students will be exploring what we are doing to the environment, and what we are doing to change the course we are currently on. Students will read and respond to news articles to help them understand current issues. Patterns and relations will be used to identify trends in flora/fauna, including population growth and decline, as well as carrying capacity of an ecosystem etc.

Project	Category	Description
Create an organism	Maker	For an ecosystem of their choice, students have to create an organism and explain how it survives in their ecosystem. They must also document their organism in a matter of their choice (news article, video, podcast)
Sustainable Ecosystem	Maker - class	Students will be be starting and maintaining the living wall in the back of the classroom. The goal

		is for us a class to create a sustainable ecosystem with flora and fauna. We will also be exploring growing food/herbs for the school cafeteria.
Eat Local	Design-based thinking	Students will problem solve how to reduce carbon emissions from food transportation. This is a broad task that will require them to understand why we don't eat enough local food before they set about making their own plan to solve the problem. I am hoping this branches out wildly, such as some students explore urban farming, others explore marketing farmer's markets, others explore creating great recipes from local food etc. The key will be for them identifying why the issue occurs.
Park creation	Maker	Students will create a park plan that functions as a sustainable ecosystem for humans and animals in the city.

Unit 3:

Science	Social	Math	ELA
Particle Theory of Matter	Ways of Life in Asia/Africa	Statistics and Probability	Novel Study

Rationale: Solids, liquids, and gas, they all play a part in our daily way of life. The same is true in asia, where living conditions are very different from ours, from transportation to food, occupations, to social issues. During this unit stats and probability will be looked at in analyzing current ways of life and quality of life in asia africa. Our novel study will be A Long Walk to Water or a similar novel detailing life in africa. In this unit the novel will serve as a base for discussions, understanding, and as a springboard for maker projects.

Project	Category	Description
Helping	Design-based thinking	With the novel students will be able to identify with what the main character struggles with on a daily basis. They will attempt to build or design something to better the characters lives in some way.
Save that water	Challenge	Design a device that will speed up/slow down the

		evaporation of water.
Phet Lab	Explore	Students will explore multiple apps on Phet lab, playing with them before coming to their own conclusions on particle theory.
Save that ice cream	Maker	Build a contraption that will keep ice cream frozen longer in the sun that your competitors.
Drink creation	Maker	Students go about making their own drinks from scratch. This will help them explore solutions/mixtures. They will be required to make an advertisement for it.

Unit 4:

Science	Social	Math	ELA
Forces and Structures	Global Quality of Life	Shape and Space	Documentaries

Rationale: Our quality of life has been affected by our ability to control the environment around us, and invent things that make our lives easier. This section will help us explore the above topics. We will be exploring documentaries based around quality of life and invention/design. Students will summarize their own and document their own building.

Project	Category	Description
Balance beam	challenge	Students will be required to get all group members balancing on a single object like a ball or cube. Alternatively they could build a device to balance two objects while exploring center of gravity.
Sturdy buildings	make	Students will make a building that is able to withstand a natural disaster. We will pour buckets of water on it, use fans on it, hail, flood etc. This will include many of the shape/space math concepts.
Making change	design-based	Look at one people’s quality of life that is being affected here or abroad. Make a change!
Human Rights	make	Make an online social media campaign promoting human rights and the way it speaks to you.

Game design	make	Create a game with structure/forces learning that details global quality of life and power/wealth

Future plans

Makerspaces take many years to build upon. The current plans for my makerspace are reasonable, however in years to come I would be looking to expand my available tools, possibly including more tech devices/robotic devices, and of course a 3D printer as prototyping is a heavy element of making and design thinking. The key is to progress your makerspace slowly. For me, the introduction of one new thing every year makes sense. For myself this year it will be Arduino; with luck the following year it will be a 3D printer!

Chapter 8 - Cross Curricular Cooperation in High School

By David Gamble

Through the interviews, research, and investigation I have done, it turns out that some of the most cutting edge skills, teaching methods, and learning experiences are present in Industrial Arts classrooms. Through a unique combination of problem based learning (PBL), project based learning (PJBL), holistic education, inquiry based learning (IBL), and community connections students learn and expand their skill set. Unfortunately, this is largely unknown outside of Industrial Arts circles and this great resource is not often capitalized on. It became clear through my research that collaboration is the missing piece to this puzzle. High school educators rarely collaborate outside a select group or their own departments.

This is not something that is done on purpose, the structure of the high school system would lead you to believe that Electronics and Chemistry are too different to possibly have anything in common, why else would we have such a rigid scheduling system? So how do you go about collaborating to create holistic learning environments that create learning opportunities that are relevant for all students regardless of educational goals? It involves communication between colleagues, teachers, and students, constant self reflection as an educator, and the drive to stay current and open to learning.

To begin the process of creating a holistic learning environment that employs PBL, PJBL, and IBL the first step is to review and understand curriculum. Curriculum must come first, this gives the educator the ability to tease out the curriculum goals from projects and assignments while keeping a holistic balance (Lux & Ray, 1970). Often the curriculum goals do not stand out at first glance when using PBL, PJBL, and IBL and because of the amount of freedom students have when using these methods often no two assignments will look the same.

The educator needs to create the guiding questions that will be used to accomplish curriculum outcomes. Even with planned guidance it is often the job of the educator and student together to identify the curricular outcomes accomplished within the student's work (Woitowicz, 2017). Provincial Curriculum is often looked at as something to teach to and the student's work must meet it, this however does not support a holistic learning environment. Instead of a student struggling to meet curricular goals, it is flipped, and the educator looks for the educational goals within the student's work (Martinez & Stager, 2013). This does not mean that curriculum should not be a major pillar of a student's education, rather it helps to make the curriculum more visible to the student and transforms it into a tool that helps to serve the student's learning and not the other way around. These methods will require educators to be masters of their provincial curriculum, giving them the ability to pinpoint outcomes from students work and make connections to the community and world around them (O'Sullivan, 1999).

Collaboration can happen between any group of subject areas or classes. Some of the most educationally valuable and incredible projects come out of the most unlikely of pairings. During the interview process amazing projects came to light that most people just do not hear about because of lack of press or promotion. For instance, did you know there is an international F1 $C0_2$ race car league where students are not only required to create a professional level CO_2 powered race car to very strict specifications in order to compete, they also must seek sponsors, create presentations, and build an original brand around their pit crews. Cars must have a professional looking paint job, all components must be built and designed by students including wheels, bushings, axles, and bodies. Students must create a minimum of two cars that are identical in all specifications so if one is broken they could still compete. The building of the car is just the beginning, beyond all of that students must seek out sponsors, design a pit crew uniform, and present their ideas to the public and judges. This project is an example of the many amazing things can happen for learners and educators. Students have voice and choice throughout the entire project from design to execution. The teacher becomes a guide or facilitator, and it goes beyond just learning about science concepts or hands on skills. The students who are successful in this program build skills traditionally found in English, science, industrial arts, business education, and many more.

Now the previous example is an amazing experience and a lot of work so depending on the teacher's and students' situations it may not be possible. This does not mean that collaborative projects are out of reach and cannot happen with a smaller time commitment or on a smaller scale. Several projects can be created and completed that allow students to have more voice and choice while keeping additional time commitment to a minimum. A great example of this is the creation of infographic posters for processes, subjects, or concepts. Students are given the choice of topic and are required to work with a teacher outside of the classroom as a client. The students are only restricted by the amount of time and resources available. As you can imagine students can become very creative while not only gaining the obvious subject area knowledge but also gain experience creating for a client. Generally the home base for this project is graphic arts or art class, but it can be spearheaded by any subject area or instructor.

As mentioned, it is the most unlikely pairings that produce some of the best educational opportunities for students and educators. Grade 11 chemistry and grade 9 electronics classes have little to do with each other at first glance, but upon closer inspection you can find a whole list of educational links that students can actively engage in to enrich their learning experiences. After many after school conversations with my local chemistry teacher we discovered that during grade 11 chemistry students need to measure solutions salinity and the equipment was very expensive for such a simple device. My grade 9 electronics students needed to learn how different electrical components worked individually and in a circuit. This lead to us forming groups of grade 11 chemistry students who could explain what they needed to accomplish in class with grade 9 students who could design and build a solution for the chemistry class. This resulted in collaboration among students, creative solutions, incredible amounts of testing and research, and a deeper understanding of content on both sides of the group. The best part of the projects was that students would freely meet outside class time to ask each other questions and tweak designs. During this project students developed teamwork skills,

Assessment can always seem like a daunting task, especially when trying to build a holistic learning environment and using PBL, PJBL, and IBL. Assessment can be used as a tool for reflection and student guidance when implemented in a holistic manner. Holistic assessment usually includes meetings between educators and students during which guided reflection is used not only to judge work but to help guide a student's learning in future projects and identifying areas of learning that need more focus and practice (Kohn, 2004). When reflection and meetings are used as check ins, student reflection will allow an educator to gently guide or help a student understand a concept, skill, or strategy. This approach also benefits the educator, making them acutely aware of individual student progress in an area and their current mental and emotional state.

While these practices are time consuming to implement they provide information not available from traditional assessment methods and can help to create a holistic environment that helps guide the development of crucial skills including critical thinking, emotional intelligence, working as a team, resiliency and more. When assessment is used as a tool to guide learning the concept of grades and percentages starts to melt away. Instead it becomes about a student's learning, not about how they can be judged and neatly categorized. Under the current system of education, it is not possible to eliminate grading and standardized testing at the high school level. Too much of the system depends on traditional assessment tools to eliminate it quickly or completely (O'Sullivan, 1999). However, it is possible to change the assessment that happens in your own classroom and to start educating teachers, administration and parents on the benefits of a holistic approach to their student's learning.

For a student to buy into a holistic classroom and gain the most amount of educational benefit from it they must have a voice and topics of study must be relevant. This can be done in several ways including the educator staying current in world affairs, technology and entertainment. This could be difficult for one person to stay up to date with all topics that are pertinent to students. An easier way is to implement a system of student voice in the classroom. This can be done through discussion, suggestion boxes, and websites. Students are often very aware of what is going on around them so by incorporating current topics educators can draw in student opinions through incorporating the topic area into the lesson or assignment (Hare, 2006). When starting out it can be difficult to tease out these ideas from students especially if it involves speaking in front of a class. The process can lead to increased trust and a change in educator-student relationship from assessor and pupil to learner and guide. With this increased level of trust, it can reduce resistance to learning from students and increase educator involvement in student learning

(Lee, Hong, & Niemi, 2014). Once the change in relationship has occurred students will be more willing to come forward in class discussions and in sharing opinions on current topics. When working with students it is important for them to recognize that their voice matters, in the educator's decisions and governing of the classroom, both as individuals and as a group.

Formulating rich questions are just as important as the topic they are based on. An improperly formulated question can result in students being led off topic or not understanding the topic that is being focused on. Vaguely formulated questions leave students uncertain of requirements and the learning objectives the educator is trying to reach (Gary, 2015). Educators should be sharing what the educational goals behind the assignments are; curriculum should not be hidden from the learners, rather it should be obvious and easily accessible. By making curriculum a visible part of an educator's teaching students will have a better understanding of the goals. This removes the curtain that it is usually hidden behind and helps to build trust between student and educator.

Questions must be focused, but not restrictive. If a student wants to do investigations related to the topic that may not have been evident the educator should let them if it will help reach the learning outcomes needed (Li, Moorman, & Dyjur, 2010). The formulation of questions must allow for student creativity and encourage them to think outside the box. Think to yourself, 'What is the real goal of my curriculum for my students?'. Is the purpose of curriculum for my students to hand in the same assignment or is it to build and foster understanding in the topic? If it is for all students to hand in the same thing the educator must be very specific what they are expecting from their students and leave as little room as possible for misinterpretation. If the goal is to build critical thinking skills and resiliency the educator should still be specific in their questioning, but let it be known what skills they are looking to develop and if a student would like to do something different that they should organize this.(Martinez & Stager, 2013).

Assignments should follow up with some form of reflection, this can be one on one meetings, journal entries, or some form of self-evaluation that requires students to do more than circle the desired grade. Reflection should not be limited to the students, the educator themselves must be open to plenty of reflection when using a holistic learning model that incorporates PBL, PJBL, and IBL (Hare , 2006). This step can be crucial when seeking out partnerships, the educator must know what they have to offer and what kinds of partnerships they should be looking for in order to complement their skills.

Partnerships when using PBL, PJBL, and IBL help to build programs and create a holistic learning environment by looking beyond the classroom. The high school education model takes related topics and walls them off into classes and departments. This can make it difficult to see connections between topic areas and how they can benefit each other. When seeking out partnerships educators must know their curriculum so they can see the connections to other areas and other specialties. It is not enough to simply know your curriculum when approaching another educator about a joint project or partnership you must come bearing ideas and thorough information about your classes' skills and what they can offer.

Educators as a whole are giving people, but they also have their own classrooms and programs to run, so simply approaching someone and asking to do a project sometime may not result in a collaboration. Instead the educators must get together and talk about what each class can offer the other and how it will benefit the students involved. At times, it is necessary to talk about a specific skill, facility, or knowledge you can offer to the other educator. By coming with ideas, knowledge and skills it makes the idea of collaboration more tempting to the other educator and more likely to succeed. Always ask if the other educator would like to become involved and never assume. Certain educators, like those from the Industrial Arts and Science departments, will often get approached with ideas and may have committed to several projects already so planning in advance is good practice (Martinez & Stager, 2013).

When wanting to form partnerships with the community and industry, educators needs to be hyper aware of how it will benefit the students. Often it is best for the educator to approach the community member or business rather than the other way around due to issues of insurance, safety, or politics (Pharand, 2017). This is done already with apprenticeship and skill based programs so consult with an administrator first before approaching or accepting an offer. A similar approach is followed with a collaboration or partnership with fellow educators, how will this benefit the students, what skills can they offer and learn, is it safe, does it aid in development of an educational goal? Classrooms are small but they are a part of a larger community and it is important for them to become involved with it (Eberhardt, 2015).

When presenting partnerships and collaborations to students it is important to do it in an exciting way, this can be done by presenting a problem and looking to them to find a solution while drawing on the expertise or skills that are now available. The class or face of the business should be present when the topic is introduced, as this will create excitement and buy in from the students involved. It can also help in the fostering of a connection between the community and the student, it allows the student to interact and understand the problem they are looking to solve (Bilyk, 2017).

To give students the best chance of success when incorporating PBL, PJBL, and IBL into a holistic classroom regular 'check in' times should be established. The student should be reflecting on their progress with the educator and other students to make sure they focus and stay on track and that progress is being made. When a student becomes stuck or headed down the wrong path it is the instinct of the educator to rescue the student and give them the answer. Instead thought inspiring questions should be asked requiring the student to reflect on the topic and learning or they should be referred to the Problem-Solving Loop or Engineering Design Loop. Educators and students alike often learn more from their mistakes than they do from their success. Allow them to attempt several different solutions until finding the final solution for the presented problem or challenge. To encourage exceptional work educators must acknowledge the student's projects will require more time than previously planned on occasion. This flexibility allows learners to explore ideas and topics more in-depth and create learning opportunities not normally available when strict dates and guidelines are applied. Educators must be careful that this is not taken advantage of and still enforce due dates for daily assignments and projects, extensions should be reserved for examples of exceptional work (Hartikainen, 2017).

Holistic education when paired with PBL, PJBL, and IBL teaching methods is a powerful teaching philosophy for the high school environment. It helps to break down the traditional boundaries of subject area and school, opening it up for community and collaboration. With a constant eye on curricular goals and outcomes and by incorporating cross curricular activities and community collaboration students get an education that more closely resembles life outside of a school system while still having the safety net of a holistic environment that helps to nurture students learning. Holistic education, PBL, PJBL, and IBL teaching methods are by no means a one size fits all philosophy for education, not every student will thrive under these conditions. Under the current model of western education, it still is not possible to eliminate standardized testing, so no classroom can fully embrace using these teaching methods exclusively, however, educators can still use holistic teaching to help their students grow academically and ready them for the world beyond education.

When creating a holistic learning environment more than just curriculum must be considered. The educator needs to identify important and current topics both in the local community and in the global community. Students need to be able to make connections on a personal level with the assignments and activities in school (Lee, Hong, & Niemi, 2014) . One of the simplest ways to do this is by creating community projects that benefit other people. Students can express their own ideas and solutions for the community while helping to build empathy and emotional connections within the student population. Completing community projects on their own is not enough, students must have a significant role in the planning and decision making process to create the authentic learning experience that helps develop emotional and empathetic connections. When a student is simply informed about what they are doing it often fails to create an authentic holistic learning experience and results in the student simply viewing it as another chore or task to finish. By involving students in the implementation of curriculum both learning and emotional goals can be met (Martinez & Stager, 2013). Encouraging student voice and involvement in the classroom and project planning process can give the educator a better understanding of their students physical and emotional health. Keeping the lines of communication open between educator and student allows the educator to guide the student towards curricular goals that may be missing while offering time for students to reflect on their learning in a meaningful way.

Cross curricular projects help to develop deeper levels of learning in the students involved. By applying skills and seeing the connection between subject areas students experience more authentic learning. Through asking rich questions, guiding students to find and solve their own problems, and facilitating and encouraging meaningful, forward thinking and self reflection students will continue to develop critical thinking skills. With the help from student voices and by using up to date and current topics educators can continue to both prepare them for post secondary education and the ever-evolving advancements in society and technology.

Bibliography

Bilyk, M. (2017, April 14). Educator. (D. Gamble, Interviewer)

Eberhardt, D. (2015, August 19). Preparing Students for Life Beyond College: A Meaning-Centered Vision for Holistic Teaching and Learning. College & Character, 197-199.

Gary, K. (2015, 9). Project-Based Learning. Computer Education, 48.

Hare , J. (2006). Towards an understanding of holistic education in the middle years of education. Journal of Research in International Education, 301-322.

Hartikainen, K. (2017, April 16). Mechanical Engineer and Educator. (D. Gamble, Interviewer)

Kohn, A. (2004). What Does It Mean To Be Well Educated. Boston, Massachusetts: Beacon Press.

Lee, D. H., Hong, H., & Niemi, H. (2014, May 15). A Contextualized Account of Holistic Education in Finland and Singapore: Implications on Singapore Educational Context. Asia-Pacific Education Resource, 871-884.

Li, Q., Moorman, L., & Dyjur, P. (2010, April). Inquiry-based learning and e-mentoring via videoconference: a study of mathematics and science learning of Canadian rural students. Educational Technology Research and Development, 58, 729-753.

Lux, G. D., & Ray, E. W. (1970). Toward a Knowledge Base for Practical Arts and Vocational Education. Theory Into Practice, 301-308.

Martinez, S. L., & Stager, G. S. (2013). Invent To Learn: Making,Tinkering, and engineering in the classroom. Torrance,CA: Constructing Modern Knowledge Press.

O'Sullivan, E. (1999). Transformative Learning Educational Vision for the 21st Century. New York: St Martin's Press.

Pharand, R. (2017, May 25). Educator and Industrial Arts Coordinator. (D. Gamble, Interviewer)

Woitowicz, D. (2017, May 24). Educator and Industrial Arts Coordinator. (D. Gamble, Interviewer)

Chapter 9 - Making and Coding in Our Connected Classrooms

By Devon Caldwell and Leah Obach

Dcaldwell@flbsd.mb.ca
http://kinderdiva.blogspot.ca/
@india0309

Lobach@pwsd.ca
http://mrsobachsclass.blogspot.ca/
@LeahO77

Introduction

The movement towards making and coding in classrooms has been strong in recent years. As educators, we have found ways to integrate these activities to create appropriate and enriching learning experiences for our early years students. From the viewpoint of experienced educators, this chapter explores how to introduce and manage coding and making activities in an early years classroom, along with sharing what early years students can gain from engaging in these types of learning activities. Additionally, we will include information on how a project-based learning approach can be used to guide students in planning their own coding and making events.

Coding

Why teach kids to code?

In addition to the ways that coding activities can help educators address curricular outcomes, coding also helps develop timeless skills such as communication, collaboration and problem solving. Students in our classrooms often work together while learning to code, which encourages the development of collaboration skills. In our classrooms, developing social competencies is an important aspect of early years education. Collaborative coding activities help students learn to take turns, support and encourage each other, provide feedback and communicate appropriately. Often, direct teaching and modeling is required for students to develop these skills. For example, young learners can become proficient in coaching a partner and giving suggestions when helping a struggling peer. Having learners tackle coding challenges together also allows children to experience the benefits of teamwork. Often, when coding, students must make more than one attempt to succeed. With appropriate support, students learn to persist in tasks that require several attempts. Coding activities encourage children to problem solve in order to correct errors in their code and move forward in the activity. Engaging learners in these types of activities helps students to become more comfortable with taking risks and helps them to learn to persist in tasks that are challenging.

How to Get Started with Coding

In our classrooms, we often start with a whole group lesson on coding. We introduce coding as "giving directions" and explain that code is what makes all of our technology work. Usually, our young learners' first introduction to coding is through unplugged coding, which involves no technology. Since giving and following directions is such an important part of early years learning, unplugged coding not only introduces coding, but also strengthens those skills. One unplugged coding activity that we use is to place an object such as a stuffed animal on the floor or 100 grid of the learning carpet. Using oral directions or cards with arrows printed on them, students are asked to give and follow directions to reach the object. This can also be done with a small object on a paper 100 chart or alphabet chart. Once students understand the unplugged coding activity, it can become a learning centre or partner activity.

Once we have introduced unplugged coding, we begin by working with a single coding tool, which is an app or website that we think will be best-suited to the abilities of our class. We usually give a basic overview of the tool, which can be done by projecting the screen of the device for all student to see. Next, we assign students to work in groups or partners to tinker and explore within that tool. As students progress, we provide individual, small-group or whole-group support in the form of feedback, mini-lessons or modeling. Sometimes, we will choose to introduce additional coding tools for students to explore. However, most tools provide levels that become more challenging and complex as the students progress, so focusing on one tool is also a good option. We'd recommend checking out Kodable, Scratch Jr. or The Foos for early years students.

Tips for Coding in the Classroom

Start with one tool. There are many free and low-cost options available for coding in the classroom. Try out a few and pick one that you'd like to share with your students, then go from there!

Use the resources available. Many great, and often free, teacher resources have been developed. Access these resources to learn more about coding and the tools that are available.

Buddy up! Find a teacher friend and work together on coding. Maybe the two of you can plan together or partner your students to learn together. Cross-grade buddies can be very effective. Connecting your classroom to another class via video calling is another way to collaborate on coding.

Use what you've got. There are coding tools available for almost any device, including web-based and app-based options. Look for tools that will work with the devices you have access to. If you do not have devices, there are many unplugged coding activities to try.

Learn with your students. Don't be afraid to learn alongside your students. It is okay if you only know a limited amount and work through it while teaching.

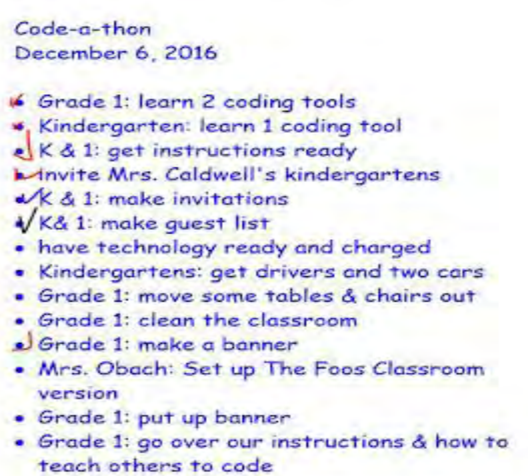
Let students be the experts. Students will quickly become experts on different aspects of coding. Watch for students who excel in coding; it may not always be the students you expect. Then, rather than always being the one to provide support, direct students to one of their peers who can support them.

Coding and Project–Based Learning

It has become an annual tradition for our students to plan a coding event to be held during Computer Science Education Week. Using a project-based learning approach, we guide students in planning and hosting their own Hour of Code™ event. Although the exact process varies each year, our general steps would include:

Introduction: We explain the Hour of Code™ movement to our students and ask them if they would like to plan an event to celebrate Hour of Code™. If students show interest, then we proceed with the next steps.

Developing an action plan: Working with students, we create a plan for how we will tackle the project of planning and hosting our coding event. Often, we use a to do list to guide us. The to do list changes as we work together to complete the project. If we are collaborating with a partner class, we typically meet via video call to stay on track with our event planning. This system of collaborative project-based learning has worked well for us to share the responsibility for planning our annual coding events.


Code-a-thon
December 6, 2016

- Grade 1: learn 2 coding tools
- Kindergarten: learn 1 coding tool
- K & 1: get instructions ready
- Invite Mrs. Caldwell's kindergartens
- K & 1: make invitations
- K& 1: make guest list
- have technology ready and charged
- Kindergartens: get drivers and two cars
- Grade 1: move some tables & chairs out
- Grade 1: clean the classroom
- Grade 1: make a banner
- Mrs. Obach: Set up The Foos Classroom version
- Grade 1: put up banner
- Grade 1: go over our instructions & how to teach others to code

Identify relevant teaching and learning: Project-based learning provides authentic opportunities for students to learn and apply skills and knowledge. Once the action plan has been created, we can identify important lessons and learning activities for students. For example, when students wanted to create coding tutorials, we used the opportunity to teach procedural writing. When students wanted to plan coding stations to be set up around the school gym, we taught them to use non-standard measurement to measure the space and then supported them in creating a map of how the space would be used.

Example from 2016: One of our jobs for this project was writing instructions for our selected coding tools. The whole class learned about procedural writing and we wrote instructions as a shared writing activity. We saved them as PDFs to print out for use at the coding event and we created a video version with narration to be played during the code-a-thon. This was a great opportunity to introduce procedural writing to my grade 1s and it always adds motivation to write for an authentic audience. Check out our final shared writing projects below:

https://www.youtube.com/watch?v=7JyFW_fLIbY#action=share

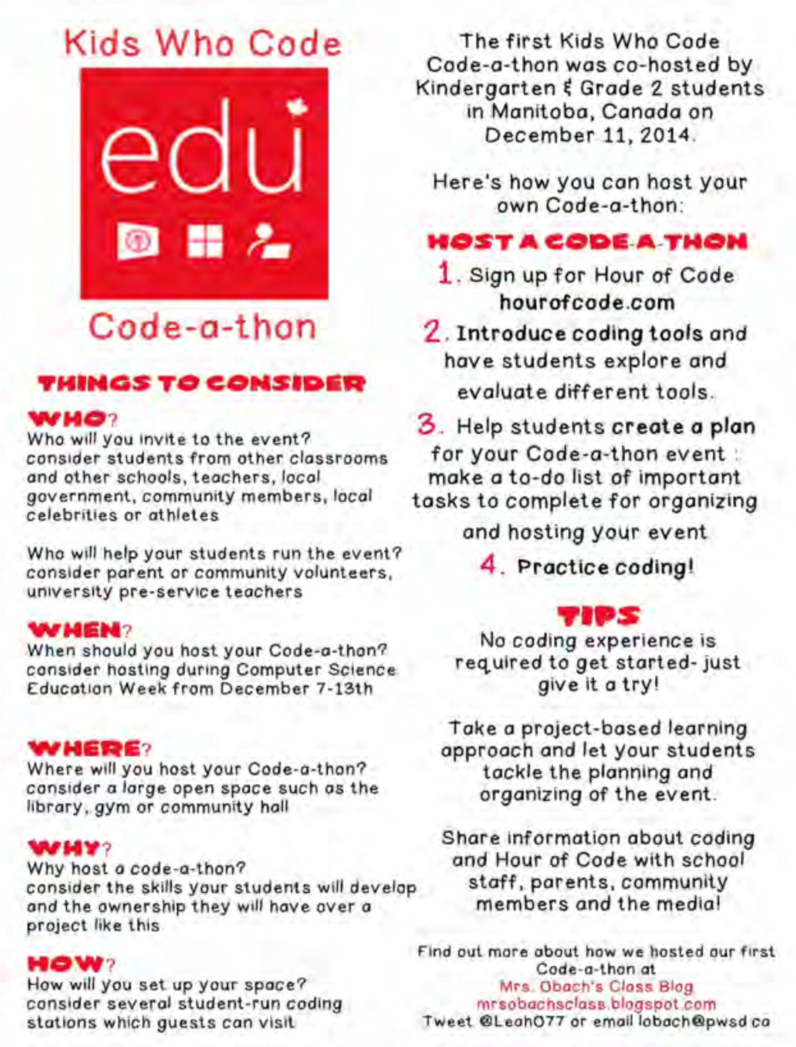
https://www.youtube.com/watch?v=aH5V8rfL2OM&feature=youtu.be

Execute the Action Plan: The to do list guides our daily planning. We set aside time in our schedule for project-based learning or integrate relevant lessons during our regular instruction times. During those times, students work together to complete tasks on our to do list, add new tasks to complete and check off the tasks that are done.

Reflection and feedback: This step is ongoing throughout the project. Both students and teachers reflect on the project, seek feedback and make changes. Sometimes this involves receiving feedback from partner classes, families or others involved in the project.

Sharing: This step is also ongoing throughout the project. Students and teachers share updates on the progress of the project, work samples, reflections and helpful resources. This sharing can take place online using social media and blogging. It can also involve sharing within the school, sharing with families and community members and sharing with the local media.

Code-a-thon Tip Sheet: Here are some tips for hosting your own code-a-thon:



Further Reading and Resources

2016 Kids Who Code Code-a-thon video
https://www.youtube.com/watch?v=STSJwYMybaU
2014 Kids Who Code Code-a-thon video
https://www.youtube.com/watch?v=wVr9tSJNyhk
Code-a-thon Tip Sheet – PDF file for download
http://tinyurl.com/ybuznn4q
Microsoft Hack the Classroom video featuring Mrs.Obach's class coding
https://youtu.be/M5nCo-Y_HJY
Mrs. Obach's Blog Posts from the Kids Who Code project
http://mrsobachsclass.blogspot.ca/search?q=Kids+Who+Code&max-results=20&by-date=true

Making

Why provide classroom time for making?

We have found that making is a wonderful way to encourage creativity, along with problem solving and design thinking. Making is a developmentally-appropriate activity for children of diverse abilities, encouraging and strengthening fine motor skills and social competencies. Maker activities provide an opportunity for students to create something without explicit directions. So often, students are expected to carefully follow directions when creating in the classroom. Maker activities encourage learners to take ownership for the planning, design and creation of a product. They have the chance to apply skills and knowledge in the process of making. Maker activities can be collaborative, providing a means to develop communication and teamwork skills. Often, students become very immersed in their maker projects and are highly engaged in the task.

How to Get Started with Making

We often introduce the idea of making by discussing with students all of the different things we can make. Through these discussions, a range of ideas emerge. Students come to realize that making includes a wide variety of activities including crafting, building, baking, construction, art and more. We work with our students to set expectations and rules for maker activities. Making is meant to be open-ended, so we keep the expectations straightforward, with a focus on safety and enjoyment. Often, these expectations are recorded on an anchor chart so that they can be easily reviewed. Next, we choose some materials that we already have access to and make them available in our classroom for makerspace activities.

Tips for Making in the Classroom

Start with What You Have. Use the materials and tools that you already have to get started with making in your classroom. There is no need to spend a lot of time or money on procuring supplies to get started.

Safety Comes First. Begin with establishing basic safety rules for using the maker materials and tools, especially if your students are using tools like scissors and staplers.

Reuse and Recycle. You can find lots of great makerspace supplies just by checking the recycling in your school or community or by asking for donations of unwanted materials.

Makerstations rather than Makerspaces. Sadly, not every school has the space to have a dedicated makerspace. Don't let that discourage you. We have had a lot of success with what we call "makerstations". Creating portable tubs of maker supplies allows you and your students to set up (and clean up) anywhere in the school. A management system for the makerstations can be a challenge. One teacher we know labels tubs of materials with red or green dots. Green materials can be used freely and red materials require teacher permission. Another system that has worked for us is storing the makerstations in a common room of the school (such as a multipurpose room or storage room) and taking out only the stations we want to use on a given day.

Provide big blocks of time. We have found the students quickly become deeply engaged in what they are making, and it is very disruptive to their creative flow to have to stop after a few minutes. Long blocks of time once or twice weekly leads to greater productivity and creativity.

Let them make anything and don't do it for them. As teachers, we are natural helpers and leaders, but this isn't the time to smother your students with assistance and rules. As long as your students aren't hurting anyone or making something inappropriate, get out of their way. Careful observation will tell you when it is time to offer a tidbit of support or advice to help students succeed.

Photograph their creations. Young students have a hard time seeing projects dismantled (Lego creations for example). We have found that it is very helpful to encourage students to snap a picture of their work. This image can be shared via social media (captioning is a great opportunity to develop early writing skills), emailed/texted to a parent, or included in a digital portfolio.

Enjoy being an observer! Don't forget to embrace the chaos and marvel at all the amazing and messy learning taking place!

Making and Project-Based Learning

We have guided our students in hosting two Maker Faire events. The first year, the event was combined with a yoga festival based on students' interests. The second year, our full day was dedicated to making. In both cases, students planned and organized the events with our support. Implementing a project-based learning approach provided many meaningful opportunities to enhance literacy and numeracy skills across the curriculum.

The following excerpt from Devon's blog explains more about using project-based learning to plan a Maker Faire with students:

Making a Maker Faire and a Yoga Festival Too: Part 1

August 12, 2016

http://kinderdiva.blogspot.ca/2016/08/making-maker-faireand-yoga-festival-too.html

Leah and I introduced the concept of making one morning in our weekly Skype call, We discussed how everyone is a maker, and shared some of the things that Leah and I like to make. Boys and girls then shared with each other what they liked to make, and we made a promise to each other to try maker space that coming week and report back to each other the following Friday. This Skype accountability with our partner class is key in ensuring that we follow through with our plans!

We assembled materials with a visit to the Co-op for cardboard boxes and scoured the supply room and the recycling bins. For added inspiration, we viewed [Caine's Arcade] on YouTube, which they absolutely loved. After a discussion of safety and scissor/stapler use, I let them loose. Take a look at the high levels of engagement, creativity, and problem-solving in this [short video]!

Very quickly, making became my young learners' favourite thing to do. At Miniota School, our buses drop off half an hour before school starts. If I wasn't in the room when my students arrived, I would find them already making, choosing to work on their creations instead of go outside and play or eat breakfast. It was impossible to contain their enthusiasm....or the mess that was taking over our classroom!

Leah's students were experiencing the same passion for making, and of course this turned into another collaborative project. Our boys and girls decided they wanted to host a maker faire where they could make things together, and teach other people about making. My students took the lead on this project, and we decided to host it in our school, as Mrs. Obach's class took the lead and hosted the Code-a-thon. As both of our classes also love yoga. we made the decision to host our maker faire on International Day of Yoga so that we could have a yoga festival in the afternoon.

As always, we followed a project-based learning model--students taking the lead, collaborating and creating with technology, and involving multiple partners. As teachers, Leah and I helped the students to "uncover" more of the curricula, providing interdisciplinary lessons as needed to help the students move the project forward. Regular Skype calls allowed us to each take on different tasks, report back to each other, and make decisions about the maker faire and International Day of Yoga Festival.

We used shared writing to compose this letter to our principal:

Dear Mr. Lewis,

We are wondering if we can use the gym on June 21st for a Maker Fair and International Day of Yoga. We would like to invite Mrs. Obach's Grade 1 class and all the kids in our school.

We have Makerspace in our classroom. Making things is good for us because:

-it helps us learn to build things

-it gives us stronger muscles when we build

-when we make things, we have to figure things out and solve problems

Yoga is good for us because:

-it helps our spine, muscles, bones, and our insides

-it makes you healthy

-you breathe better

-it makes you calm, flexible, and happier

We would like to do our Maker Fair in the morning and yoga activities in the afternoon. We would invite the other classrooms to sign up to join us.

We love you Mr. Lewis and we hope you say yes!

Our next job was to create invitations. We used Microsoft Publisher to make these simple cards, and also exported them as a PDF to email to faraway guests. QR codes on the back of the invitations provided links to movies and more information. We made an extensive guest list and addressed and decorated the envelopes. We used tally marks to see how many stamps we would need to buy and how many could be hand-delivered.

Miniota Elementary School Kindergarten and Hamiota Elementary School Grade 1 classes are planning the first-ever Maker Faire and International Day of Yoga Festival!

What is a Maker Faire?

A Maker Faire is when a whole bunch of people come together to make, create, invent, and solve problems. We use things like cardboard boxes, tape, string, staplers, wool, craft supplies, recycled items, and technology. Making is good for kids and grown ups because it helps us be creative, work with others, and solve problems.

What is International Day of Yoga?

The United Nations declared June 21st as International Day of Yoga. We practice yoga to help our bodies get strong and flexible and to feel calm and happy. Yoga helps us focus, and this makes it easier to learn at school. Anyone can do yoga!

Tuesday, June 21, 2016

Miniota School

Opening Ceremonies: 9:45 am

Maker Faire: 10 am-12 noon

Join us to make, invent, and create using a lot of different materials!

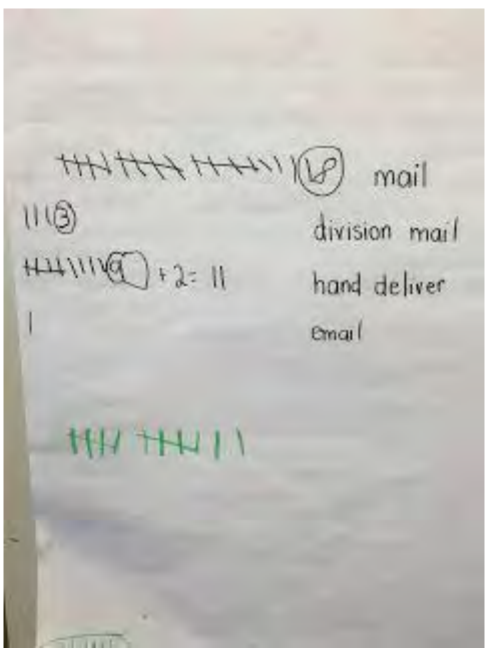
International Day of Yoga Festival

Yoga activities: 1-2 pm

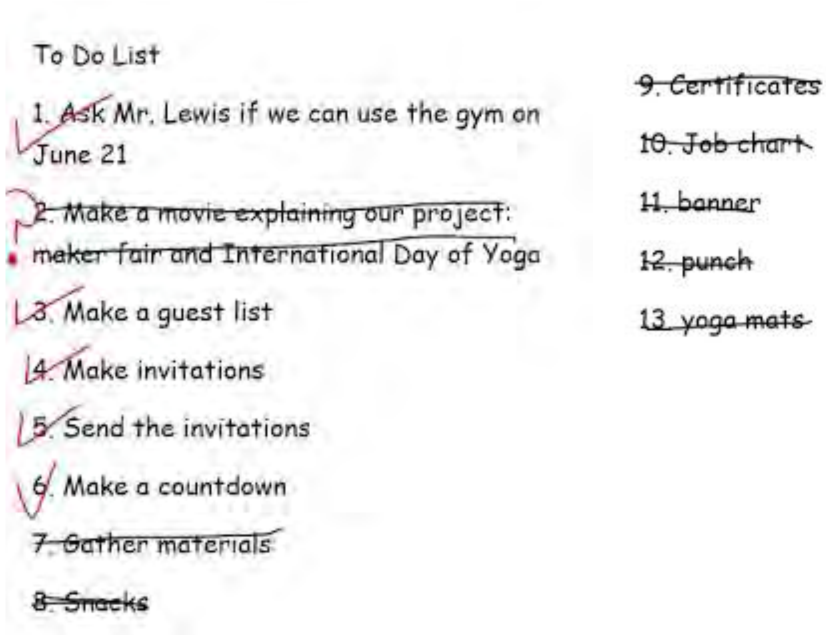
Yoga Class: 2-2:30 pm

Everyone is welcome. Please bring a mat.

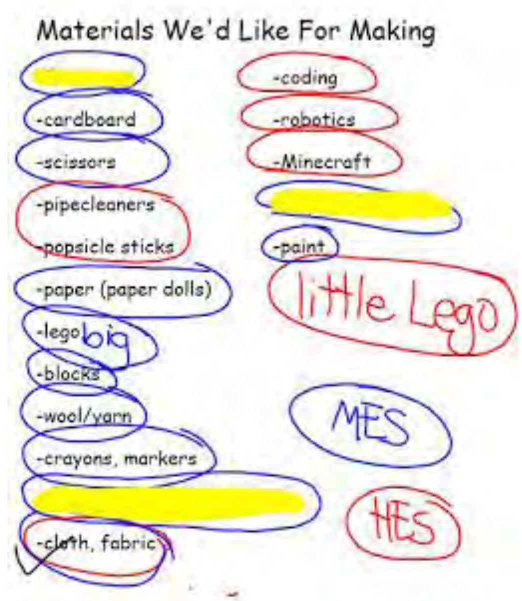
You can drop in at anytime and see what we are doing, or join us for the opening ceremonies or the yoga class.



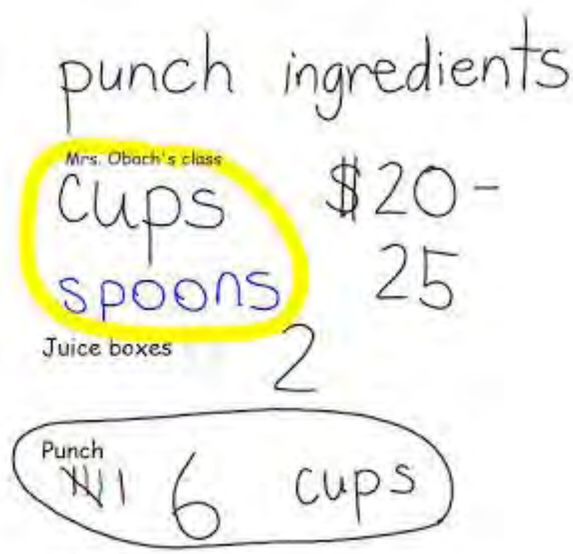
As for every project a to do list guides us and acts as a plan.



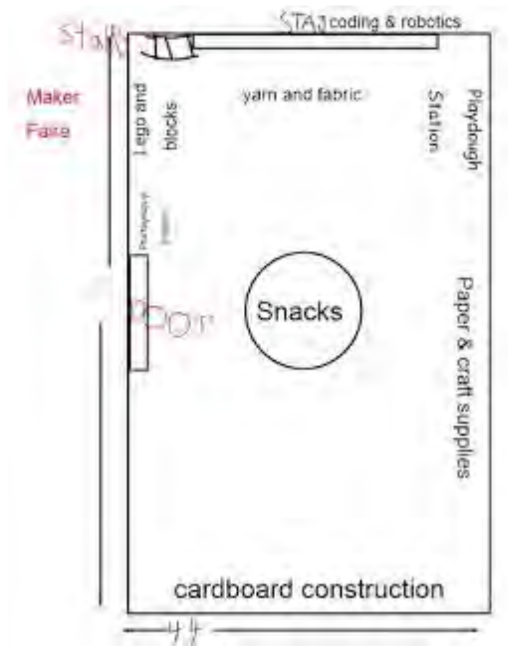
With our Grade 1 friends, we brainstormed a list of materials that we wanted for making. We divided up the list, making decisions about who was responsible for getting the items. We also put out a plea on social media for donations.



We collaboratively developed this schedule for the day and a list of healthy snacks. We decided to ask our parents to donate snacks, and they were wonderful to provide everything we needed. We asked Mr. Lewis (our principal) for a budget to buy ingredients to make punch. This led to some excellent math learning. We walked to the Co-op and shopped for supplies.



One of the activities I was most excited about was mapping the gym, as it brought in social studies, numeracy, and ELA outcomes in a very authentic way. We paced out the gym, counting our steps, then drew and labelled this map on the SMART Board. This map was later posted in the gym to help us set up.



Developing a job chart was much anticipated by the boys and girls! They decided that the boys would act as greeters at the door and the girls would preside over the guest book and snack table. We were also lucky enough to enlist the help of our Grades 7/8 class and Hamiota Collegiate student council.

Further Reading and Resources

Blog post: 2016 Maker Faire and Yoga Festival

http://mrsobachsclass.blogspot.ca/2016/06/2016-maker-faire-yoga-festival.html

Blog post: Makerstations: Portable Makerspace Activities

http://mrsobachsclass.blogspot.ca/2017/01/makerstations-portable-makerspace.html

Renovated Learning blog by Diana Rendina - http://renovatedlearning.com/

Krissy Venosdale's blog - http://krissyvenosdale.com/

Conclusion

We have found that coding and making activities can lead to rich learning in our early years classrooms. Not only do the activities themselves build important skills, but coding and making also provide opportunities for interdisciplinary learning and project-based learning. Both coding and making activities are well-suited to student-directed learning, so teachers ready to try these activities should be prepared to give students choices and provide support and guidance as needed. Setting clear expectations, modeling appropriate behaviours and delivering relevant and timely lessons are important to ensure coding and making are positive learning activities in the classroom. If you are interested in introducing coding or making in your classroom, look to the wealth of online resources for ideas to get started! The lists of resources shared above would be a place to begin exploring these topics.

Devon Caldwell is a kindergarten and special education teacher from Kenton, Manitoba. Currently on leave, she is pursuing a PhD in Curriculum and Instruction (Early Childhood Education) at University of Victoria. In her practice, Devon strives to provide timely early intervention and developmentally-appropriate programs for young learners. She is passionate about infusing technology to foster collaboration, inclusion, and wellness among students, teachers, and their local and global communities. Devon believes that with the right tools and guidance, young children can make a big difference in our world.

In 2009, Devon graduated from Brandon University with a Master of Education degree in special education. Devon has been designated a Microsoft Innovative Expert Fellow and she is a proud recipient of the Canadian Prime Minister's Award for Teaching Excellence, Early Years Teaching Excellence Award from her district, and ManACE Educator of the Year Award. Devon is an enthusiastic contributor to online professional learning communities and loves mentoring educators and pre-service teachers in her role as a sessional instructor at Brandon University. Devon is also a certified yoga teacher, and when she is not at school, you can find her on her yoga mat or travelling the world. Learn more about Devon on Twitter/Instagram @india0309 and explore her blog Kindergarten Diva.

Leah Obach is an early years teacher from rural Manitoba. She is passionate about improving education through strong classroom practice, presentations at education conferences and mentoring other educators. She strives to engage her students in meaningful learning experiences, while fostering the development of important 21st century skills. Project-based learning, technology infusion and cross-classroom collaboration are important components of her classroom teaching. Leah has been recognized for her commitment to best practices in teaching. Most notably, she was selected three times to represent Canada as a Microsoft Innovative Expert Educator and, as an alumna of the program, she is now designated a Microsoft Expert Educator Fellow. Follow her on Twitter @LeahO77 or read more on her classroom blog mrsobachsclass.blogspot.com

Chapter 10 - Growing A Technology Model from K - 12

By Kirstan Osborne and Elizabeth LaPage

The technology program at Balmoral Hall is evolving ~~to~~ in order to allow girls to be risk takers, computational thinkers, tinkerers, and creators of technology rather than merely consumers. In this section, Kirstan Osborne and Elizabeth LaPage will share the changes and growth in the technology program.

Two years ago, I (Kirstan) made something of a bonkers decision. After spending ten years as an elementary classroom teacher, I saw an intriguing internal job posting, and felt an irrational, magnetic compulsion to apply. The position was that of Middle School ICT teacher & Junior School technology integration coach, here at Balmoral Hall School for Girls in Winnipeg. What sparked this strange and captivating pull to teach an age group I had never worked with before, in a field where I had pretty rudimentary expertise? Well, as I planned to pull out the Grade Five weather unit for the sixth year in a row, I had a moment of excitation in reaElizabething that if I were to move into the area of technology, my job would never be boring again! There would always be a new challenge – something new to learn or apply to the curriculum. Technology is ever changing – the position would never stagnate.

Secondly, the way I envisioned teaching ICT really fit with my teaching philosophy. What I had loved most about teaching Junior School was using authentic, inquiry-based teaching methods. I felt that the core principal of student led learning in inquiry, lent itself particularly well to ICT. Therefore, I saw my lack of expertise in the subject area as an asset, rather than a deficiency. My hope was that I could frame and facilitate projects while learning along with the students as we worked our way through them, and acquire the knowledge and skills to help them as time went by. Though it must be said, I also felt strongly that this subject was one where some students would know more than their teacher, and take their projects beyond expectations. Fortunately, I was OK with that – I had to be!

Prolific Ed-tech guru, A.J. Juliani (2017) says "Our job is not to prepare students for something. Our job is to help students prepare themselves for anything". I wholeheartedly concur. How can we prepare students for work in a world where we cannot tell what the job market will look like even ten years into the future? The way I envisioned teaching ICT was a way where skills important to any job, to *life* in fact, would be emphasized. Perseverance, flexibility, problem solving, critical thinking and ideation, would be underscored.

My final reason in making the absurd decision to apply for this position was my belief in having our girls understand that they can be creators, rather than passive consumers of technology. Quite clearly, from the hallways filled with girls whose noses were glued to their smartphones, this was not yet the pervasive belief amongst the school population. I'm quite sure the thumbs deftly poking away at the screens, were not doing much coding! Combined with this, the idea of promoting the possibility of a career in ICT or computer science; doing **something** to break down gender barriers, felt like important and fulfilling work.

With nobody more surprised than me, I was offered the position, and with more than a little trepidation, I began my ed tech journey. Luckily for me, shortly afterwards, my fabulous colleague and technology spirit guide, Elizabeth LaPage was hired to the position of senior school computer science teacher. Together we set about navigating the big picture of computer science in a N-12 all-girls school, and making the technology program our own.

Junior School

So, what does this all look like on a Junior School level? It starts by making coding hands on and tangible for little ones. To begin with, I was bedazzled by the array of apps and robots available for the early years market, and my initial thought was to use these tools to push coding with the primary kids. However, after a PD session that Elizabeth and I gave on the topic, my colleague, Kindergarten teacher, Riley West, was inspired to go on a Google search to find the best way to integrate programming into her curriculum. This led her to a video posted by one Pana Asavavatana, a KA-2 Technology & Design Coach, based out of Taipei, Taiwan. The video documented Pana's final COETAIL project, featuring an explanation of an unplugged coding unit she had developed for very young learners, with an added bonus of a global coding exchange. After watching this video together, Riley and I felt that we had discovered electricity simultaneously, and we used Pana's unit as a base for our very first Kindergarten coding project.

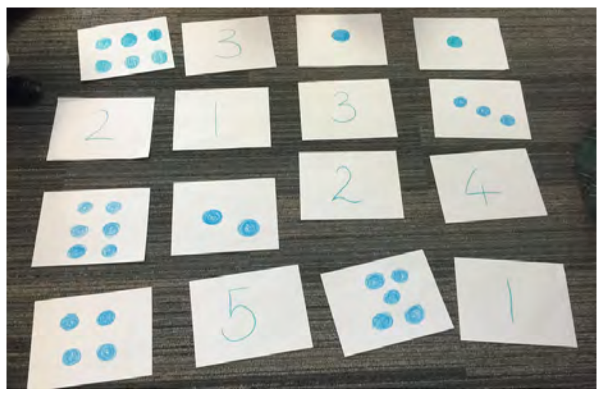
We started by showing the girls coding videos by DK Books, and together figured out that computers are all around us, and are used for many things. They can be cars, washing machines, coffee makers, airplanes. We next learned that computers are essentially empty boxes until a programmer writes code to make them do what we want. We learned that there are many different kinds of computer languages, and that we need to give a computer very clear step by step instructions to make it work efficiently. We learned that this set of instructions is called "an algorithm". We practiced thinking of algorithms for washing our hands, and brushing our teeth. Once we were clear on these fundamentals, we were ready to learn our coding commands. I explained we would be coding each other, as if we were robots ourselves. The basic coding commands that the girls were given, required them to use their bodies to show directions, as seen below with some lovely older girls acted as our models!

After running through the commands a quite a few times, I showed the children a number of coloured squares laid out on the floor, and challenged them to "code" their friend through the puzzle. The "robot" could only step on a particular colour (in the photo below, purple) from start to finish.

While the children were enthusiastically making their body shapes, we also wrote arrows on the board as another way of expressing the algorithm. The hardest part, we found, for the children, was turning. Firstly, directionality was a little challenging, and secondly, remembering to ONLY turn when given the turn command was tricky. This was our first lesson in perseverance. We had the girls strategize by facing the same direction as the robot, and take a little time to think about which way she should go before giving the command. We also emphasized the importance of asking a friend for help to solve the problem. Generally in school we want the students to work independently and show their skills autonomously, however, in computer science, we want to show that programmers work as teams. In addition, we want girls to overcome obstacles, rather than give up; and collaborating with a classmate over a problem is one strategy that is helpful to internalize from a young age.

After these initial colour patterns, we were off to the races. It was so clear from then on forward that coding could fit in with any area of the curriculum. It didn't have to be some abstract and scary concept – anyone could do it!

In this first unit, we stuck with the floor coding puzzles and had the girls write and decorate –et and –at sound words to code themselves through. We made math puzzles where we needed to code our robots through paths that added to ten. We coded our way through the story of "The Gruffalo" by Julia Donaldson.



As we journeyed on, opportunities naturally arose to teach the concept of debugging. Sometimes we debugged on the spot, sometimes we waited until everyone had added a command before "running" the program and realizing we needed to make a change to the code. We experimented with figuring out the most efficient way to code our robots through the puzzles. Lastly, the girls were invited to create their own coding commands which was rather wonderful. We had jumping, spinning, blasting, destroying and then we had the children create their own coding puzzle to share with the Junior School at their assembly on sustainability. To connect with this unit of inquiry, the children made a puzzle which featured pictures of items which were "good" for the planet, and pictures of items which were "bad" for the planet. The point of the puzzle was to "blast" all the bad things off of the floor plan. If you landed on a power station for example, the body command of spinning quickly in a circle meant that the power station would be taken from the puzzle because the students had decided that that meant metaphorically, the Earth would be better off.

It was really amazing to see such enthusiasm and learning from such empowered young girls, and it was great to engage the whole of Junior School in computer science in one fell swoop! Once this initial coding unit was conceived of, the rest of the junior school program began to fall into place. In addition, Elizabeth and I had spent time researching ways to have the computer science program follow a logical and sequential progression from Kindergarten to Grade 12. Using the K-12 computer science framework, developed by a number of significant stakeholders in the computer science field, we were able to identify the concepts we wanted to cover at particular grade levels and develop age appropriate lesson plans to introduce them. With the very little girls, this generally goes back to hands on, fun activities. Dancing to teach loops, games to introduce event handlers and so on. Those fancy apps and robots come in a bit later to keep engagement going, and then we work more formally on projects tied to the general curriculum as we get into the upper elementary grades. At the moment, we are working on integrating French and Computer Science by developing a digital storytelling task using Scratch which connects with the AIM program.

The beauty of the Junior School programming is that I can schedule as much time with teachers as we need, so we get a really good continuity going on a project. Is the goal still to work towards more coaching and less teaching? Always. That is something that requires work, both in terms of my own comfort level with the coaching cycle, as well as understanding from the Junior School teachers of what my position is destined to become. Like everything, it is a work in progress.

Middle School

What does the program look like as we continue on through Grades 6 – 8? At these grade levels the girls come to ICT twice per eight day cycle, which is definitely not enough time! If there are special events in the school day, sometimes I don't even see the girls that often. At times, this does not make for great continuity. Nevertheless, there are several large units in the middle school which are inquiry based in nature, and at this age, the girls are more able to interact and shape the planning of the projects alongside me.

Perhaps the best example of this is a Grade 8 unit named "Innovate!". The crux of this unit is the essential question *How can I use technology to create a useful product?* The girls are challenged to create a product which solves a problem. This can be a fairly daunting thing for the students, so we discuss the fact that we do not need to solve world peace; it can be a small problem that the girls identify in their everyday lives. We also watch some Wallace and Gromit videos to break the tension and show that our products could be fun as well as useful.

The unit begins with a tour of a variety of resources we have available to us, a brief explanation of how they work, and then an invitation to explore them over three classes. There are many tools at our disposal; Littlebits, Spheros, Makeymakeys, 3D printers, and a myriad of building and crafting items. If the girls want to go the software route they are introduced to Scratch coding, HTML/CSS/Javascript coding for website building and the MIT app inventor for app building. Following the exploration of materials, the girls are expected to come to some sort of decision on the item they will create.

At this point we stop, and we create a project planner to help focus and hone our ideas. We also look at the Empathic design cycle in order to think about the importance of the hypothetical user of the product and their wants and needs. This involves some research and thought with the main point being that the customer be front and centre in the design process. Following this, the girls are asked to frame their project with a reasonable timeline. We have an end date in mind, and we figure out what we think we need to do to meet the goal. During this discussion period, we also think about how we can best chart and assess our progress. When the students are part of the decision-making process, it means they have more ownership over the project and are more likely to work hard on it. The final summative assessment of this unit is a "Dragon's Den" style presentation with the peer groups giving each other critical feedback.

For the most part, the girls are left to their own devices to work through the design process to bring their projects to fruition. Of course, I am always there to answer questions (if I can!) and provide guidance, but the girls are encouraged to use their own problem-solving skills to reach their goal. Naturally, as the unit progresses, girls encounter problems. For many, it takes a great resilience to stay on the path to achieving the creation of a prototype. Naturally, many girls want to take the easy route out and change projects all together – choosing a much simpler task. Many ask what they **must** to do to get a good grade - "Will I still get a 4 if I do "x"? This is precisely what we are working to change. We want the girls to think in terms of "learning" rather than "grading", and looking at "failure" as "learning experience". To this end, discussions frequently take place about approaching tasks with grit and persistence – a "growth mindset". In addition, we have filled the labs with "can – do" text images which are intended to motivate and inspire the girls, as well as infographics about female role models in technology. Seeing the girls staying the course, and finally getting an app, website or prototype to work, is so rewarding, and the look of joy on the girls' faces is amazing! Or when the line of soap dishes she had been 3d printing suddenly turns into a line of hair brushes instead, and the project takes a whole different direction but she still stays the distance - that’s rewarding. When students take themselves to coding workshops or makeymakey workshops outside of the class time, that is sure evidence of deep engagement. Each project holds in its design the potential to make someone's life better.

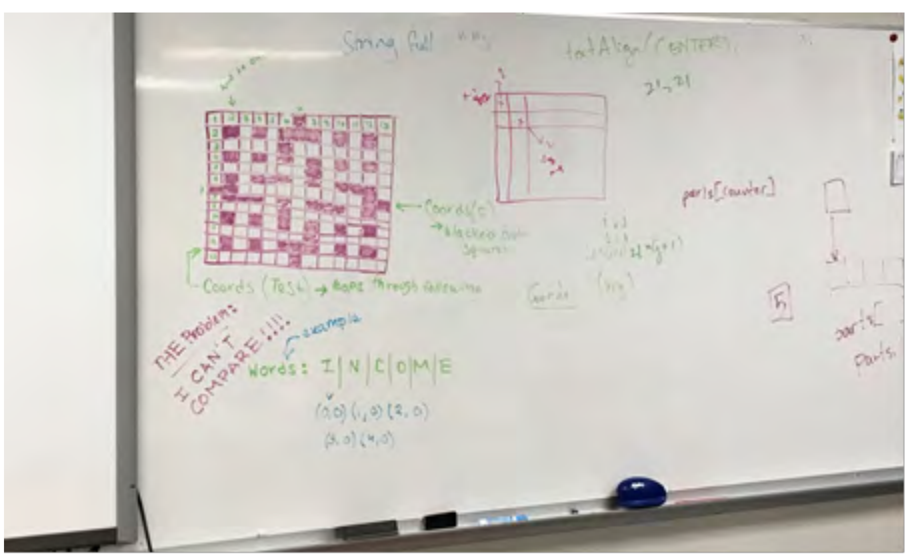
Amy's App to teach sign language

Is it a hairbrush or a soap dish?

3D Printed Guinea Pig food sphero chariot

Senior School (Elizabeth)

It’s mid-May in the computer lab. Students are filing in for their 30S Computer Science class, pulling out laptops, a steady stream of conversation happening. Some of it is normal small talk - what they have been watching on Netflix, their worries about the upcoming science quiz, the hilarious thing they received on Snapchat earlier. As the class goes on, intermingled in the conversation are exclamations of excitement, groans of disappointment, and heated discussions as students hash out coding problems. These are the sounds of students in the middle of their final projects, a display of what they have learned throughout the year and how creatively they can apply it. The final project is also a good representation of the changes that are present in the high school computer science program at Balmoral Hall.



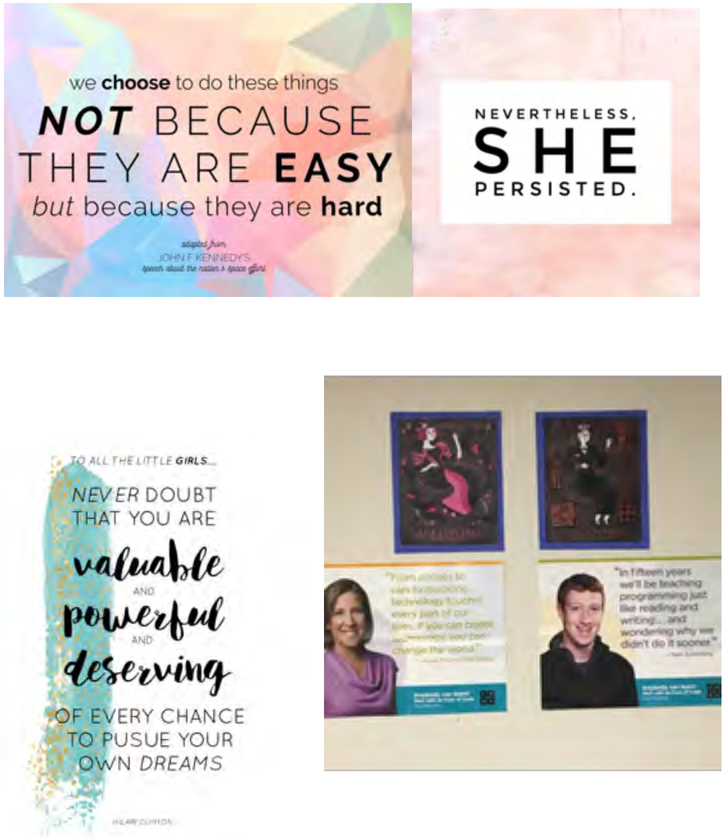
A student mapping out their program in attempt to find their problem.

The final project is adapted from Larry Wachs at Sturgeon Heights Collegiate. It is a comprehensive project that essentially challenges students to display their best work. The students are put through a thorough planning and reflective stage before and after their project, respectively. The students are not penalized for an unfinished product, rather encouraged to reflect on the whys and next steps. The final project encapsulates what the course has become: an environment where risk-taking is encouraged, failure is accepted as part of the process, and a focus on the work rather than the final grade attached.

Learning to Leap

"Most girls are taught to avoid risk," Reshma Saujani points out in her TedTalk *Teach girls bravery not perfections* (Ted, 2016). I came to understand this statement as a truth at Balmoral Hall. By the time the girls get to a formalized computer science class in grade 9, they want to make sure they do things right. Coming from a different school where the vast majority of my computer science class were boys, I was unprepared for the amount of trepidation and even tears that I would encounter in my classroom. There is no possible way to learn computer science without being comfortable with the idea that something may not work or be the best and only solution. It was clear that the lab had to be viewed as one that was safe and collegial, and there were some changes that could be made. The physical presentation of the room had to be a welcoming and motivational environment, the programming language that I used could be improved, and the nature and attitude towards assessment had to be one that did not punish students for not "succeeding".

Changing the physical classroom is an ongoing process. The first step was to add artwork that created a fun, inspiring and motivational environment for everyone. The University of Washington published research that claimed that making the classroom "less geeky" would make more girls welcome in the computer science classroom (McElroy, 2017). As a geek, I certainly did not want to get rid of every geeky reference that I had, so I still have some fun, techie posters on the wall. I feel like the students that do identify as the stereotypical nerd computer scientist still need to feel welcome in the lab, so while there's a limit to those posters, they still exist on the walls. A larger focus has been on computer science education posters, such as those from code.org, as well as images of prominent women in computer science and engineering. There are also a variety of motivational posters, particularly those targeted towards girls. A combination of all of these will hopefully allow all girls taking the class feel inspired and motivated in the space.



Some samples of the artwork and posters on the wall in the lab

The space is also changing in the technology that you can see in the classroom. When it was time to invest in new computers, we decided to purchase laptops. This created a far more open space rather than girls having to peer over large monitors. It also allows girls to move around the room more freely.

The next step that I would like to take in the physical atmosphere in the room is to increase an environment of tinkering. The school has started a robotics team, so the table and field are set up in the classroom. The table takes a lot of space, but having it there brings interest and a focus on the other areas of computer science. Slowly but surely, I am also creating an "electronics station", with bits and components, Arduinos and Raspberry Pis. These stations are not only peak interest but allow girls to become comfortable with different aspects of computer science and engineering.

The physical environment is slowly turning into one where trial and error and play are commonplace. The next step to take was to change how the content was delivered. All good computer science teachers know that the language taught in is not the focus in computer science education. The concepts and computational thinking are the important aspects, and how they are delivered changes as better options are created. For the longest time, I was teaching these concepts using Java, both in a purely textual-based context, as well as a graphical user interface (GUI) context. While the GUI assignments are more fun than compiling and running in the terminal, there is still quite a bit of overhead, especially when dealing with graphics. After attending the University of Manitoba's Computer Science Day talk on Processing, I saw that this was a language that would fit everyone's needs. Its background is Java, so students would still have a good foundation for AP Computer Science A. Relatively complex concepts such as object orientation and recursion can be learned using Processing. At the same time, coding graphics with Processing is incredibly easy to start, with little overhead; students can start coding and making something cool right away. It is easy to see results, particularly positive and fun ones, immediately. Switching over to Processing has definitely made students more willing to try things, more confident that they could get things, and more willing to try and try again.

Finally, the way I was assessing the girls has been changing and adapting so that they could focus on the concepts rather than getting a grade. I had been changing my practices for years, starting at the school I had previously taught. In my first few years, I had been marking assignments like they were math assignments – a point given or taken away for a certain number of errors, a point given or taken away for an erroneous conditional statement, et cetera. This practice already troubled me. I had started using rubrics, and have been adapting them so that they best reflect what I am looking for at each level. My assessment practice further changed when I arrived at Balmoral Hall. They were starting a practice of issuing formative and summative assessments, and allowing the formative assessments to not count towards a final grade. I saw this practice fitting in with how I wanted computer science to be assessed, particularly at the beginner level. I generally assign exercises for students to try, small programs that focus on one main concept. Students are given time to try the program without the stress of a grade, and at the end of that period would be given feedback that they can focus on rather than a grade that they will ultimately forget. With practice and feedback, the students are confident attempting more complex problems.

Finally, the general attitude that I give hopefully helps students feel like they can take a risk. I feel like most of my time spent during class is as a motivational speaker or a cheerleader, telling the girls that, "yes they can!" I will be interested to see how this role will change, as more students filter through Kirstan Osborne's courses. In her inquiry-based courses, students are already encouraged to explore, discover, and tinker. The majority of the girls will be accustomed to taking risks. They'll be acclimatized to not fear when something does not work but instead appreciates the journey that they were on. I look forward to see projects and ideas when students are already prepared to take a risk.

Rethinking Failure

Failure is a fairly scary word, especially to students. To many, “failure” actually means “not good enough,” particularly to young girls. Young women will often take mistakes personally, whereas young men do not. As Reshma Saujani pointed out in her Ted Talk, young men will ask “what’s wrong with my code,” whereas young women will ask, “what’s wrong with me” (Saujani, 2016). A part of risk taking is understanding that failure is what you make it. To many of my students, success means getting the program to work exactly the way they want it. Anything less is failure. Any computer science educator knows that there is so much more when writing a program: the problem solving techniques, the planning, the design, and the concepts that went behind most of the program. We know that one small line, or one logical error, can be the difference between a perfectly working program and one that does not. So part of getting girls comfortable with risk taking is understanding that what they think is failure is just another step in the process.

A basic problem solving tactic in computer science is breaking down problems into achievable tasks. I do this often with my students, showing the different steps that need to be done, starting with the easiest, going up to the challenging. Many students gleefully call out, “Check!” when they have completed a task. Of course, breaking down a problem is required in computer science, but it also helps students to see the value of different parts of the process. They can see the value of each step, and that when a program does not work in its entirety, it is only because of a missing or broken part.

For their computer science 30S final project, one of my students decided to make a crossword puzzle. It required file-reading, String processing, arrays, and objects. She learned two-dimensional arrays by herself and committed herself to making a well-written program. It was a very ambitious project, given the time, and it was hard for her when she realized that perhaps she was not going to get it completed. She is an incredibly high-achieving young woman who expects much of herself. We spent most of her grade 9 (computer science 20S) year working on risk-taking and accepting “failure”. This project was truly her test. It did take time, but she came to realize that the amount of work that she had done thus far were amazing accomplishments on their own. While I was proud of her work, the best part was that she became proud of her work. A year before she would have called the project a failure; now, she is able to rebrand that as a process. Her mindset is one that I always try to help the girls work towards. Failure is a part of the process in computer science, but students cannot be afraid of it.

Assessing For Learning

My assessment process has radically changed, from marking like a math teacher to a more formative-centric approach. I had mentioned that this change in assessment helped the girls take risks because there was no mark attached to exercises and therefore no reason to be worried about doing poorly. This approach is not about making things easier, but shifting the focus. In a university preparatory school, it is hard to get away from grades being the main focus. No matter how interested in the subject matter a student is, they still want to do well. To help them achieve their best, I usually require the students to choose to write one program out of a choice of a several in their summative assessments. Some of the questions are easier, mirror the exercises closely, and shows that the students are proficient. The problems get progressively harder, allowing the students who want to take the risk to give it a try. Rubrics are broken up into three parts: knowledge and understanding, problem solving and design, and standards. The knowledge and understanding assess the concepts – the student is able to use an if statement or a loop. The design portion has to do with how they approach the problem, and how well they use the different structures. Students know that they can do well on all portions if they choose a problem that fits with what they are interested in. Having a choice allows students the ability to show how well they understand the content.

My favourite assessment of the year is the final project. As described before, it allows students to determine their own project. I get the chance to see the student's creativity, their thought process, and their ability to be okay with failure. During the planning process, they must come up with levels of functionality, which gives them goals to reach. Most of the students come up with a program that they are interested in. They do not have a finished program that they have to achieve, just a goal and their personal honor at stake. They are all invested in the projects, and work hard to get things working. Even if they do not achieve what they wanted to, they still have a final reflection to talk about where their problems lie and what their next step would have been if they had more time. They have that safety mat to allow them to try a project that interests them, that allows them to take a risk. This project gets students to worry about the project, not necessarily the grade that they achieve.

I cannot take away all grades, and I cannot get the students to not entirely care. The summative assessments and grading practice, however, can be adjusted to allow the students to be prouder of the working project that they have rather than the grade attached.

Our goal for this program is to give girls the opportunity of seeing themselves as inventors and producers of technology, to see code as another way of expressing themselves, and, most importantly, to be risk-takers. We are excited to see how it will continue to grow!

Works Cited

McElroy, M. (2017). *To get girls more interested in computer science, make classrooms less 'geeky' | UW News*. [Online] Washington.edu. Available at: http://www.washington.edu/news/2015/08/24/to-get-girls-more-interested-in-computer-science-make-classrooms-less-geeky/ [Accessed 11 Oct. 2017].

Juliani, A.J. (2017). What happens when students embrace design thinking? [web log comment]. Retrieved from http://ajjuliani.com/happens-students-embrace-design-thinking/

Saujani, Reshma (2016). *Teach girls bravery, not perfection*. [Video] Available at: https://www.ted.com/talks/reshma_saujani_teach_girls_bravery_not_perfection [Accessed 11 Oct. 2017].

Kirstan Osborne
ICT teacher/Tech Integration Coach
Balmoral Hall School

Twitter: @wpgteacher77

This will be Elizabeth's 8th year teaching, and she has spent that time helping students realize that scrawling code in the margins of their page is their own form of doodling. She loves being a part of the technology education circle, from being a part of the Manitoba chapter of the Computer Science Teacher Association to leading several sessions for the Winnipeg chapter of Ladies Learning Code. Currently, Elizabeth's greatest joy is seeing the young women she teaches at Balmoral Hall exclaim in excitement when they have finally gotten their program to do exactly what it is they want it to do.

Email: elapage@balmoralhall.ca
Twitter: @erlapage

Chapter 11 - Should We Teach Our Students Computer Science?

By Eleni Galatsanou Tellidis

This chapter is an updated version of the article "Teaching Computer Science and Coding in Schools" first published on The Manitoba Association for Computing Educators' Journal in April 2017.

September 2017

Computer Science Education has become a popular topic in the discussion regarding educating our students for the 21st century. Over the past years there has been an increasing advocacy on why we should teach computer science in schools and on "teaching kids how to code". Code.org, Code Academy, Canada Learning Code are a few out of many examples of non-for-profit organizations who have emerged over the years and have been advocating about the benefits of teaching children computer science and coding. Most of these organizations are not just advocates. They provide the tools and resources for students and teachers to pursue it. Most recently, big companies like Google and Apple have launched their own initiatives to support this trend. In addition, more and more educational robot kits (e.g. Sphero, Ollie, Ozobot, Root), are becoming available that allow teaching coding and robotics in a fun, engaging way.

Politicians have also been on-board with the trend, which underlines the economic importance of teaching computer science in schools. In the US former president Barack Obama (2013), and in Canada most recently Prime Minister Justin Trudeau (2016) have been finding opportunities to highlight the benefits of having pupils (and adults) learning coding and computer science. Provinces such as British Columbia and Nova Scotia have recently committed to prioritize coding in schools as part of a broader policy to support their province's tech sector (Frelix, 2016). In British Columbia specifically, starting September of 2018, grade 6 to grade 9 students will have to complete one coding course as part of the Applied Design, Skills and Technologies Curriculum. The province supports this new initiative with significant investments in teacher training, equipment, and resources (Wilson, 2017). Overseas, countries like Estonia and the UK introduced coding in primary schools in 2012 and 2014 (Sterling, 2015). The underlying idea among all these initiatives is that every child should have the opportunity to learn computer science and coding in schools, so our children are better prepared for the fast-growing high-tech world in the 21st century.

What is the difference between computer science and coding though? Coding or programming is an important tool for computer science, but computer science is a broader field that engages mathematical concepts such as computational thinking, logic, and problem solving. Coding, on the other hand, is a tool for computer science to create software. Ellis and Corcos (2015) illustrate the difference between coding and computer science with this example: "In programming you generate a random number with the function Math.Random(). In computer science, you learn how to build algorithms that make truly random numbers that can be used in a function like Math.random() to generate a random number" (para.11). The computer science curricula 2013 for Undergraduate Degree Programs in Computer Science (ACM-IEEE, 2013) identifies 18 Knowledge Areas, which constitute the basis of essential knowledge a computer scientist should have. Programming Languages was only one of those, while some of the other areas included were Algorithms and Complexity, Architecture and Organization, Information Management, and Social Issues and Professional Practice.

Despite the prevalence of computers and their application in our daily lives, there is no universal consensus on why and how we should teach computer science (Carroll, 2015), and whether children in schools should learn how to program a computer or should focus on learning the fundamental concepts underlying computer science. Franklin (2015) highlights the need for computing education research which will shade light on how students learn computing concepts, and on what are the best methods for teaching them. "Computer science instruction suffers from a gap in content, methods and tools" Franklin (2015, p.35) argues, and she proposes that computer science departments take the lead in collaboration with education departments to assist in this direction (Yadav et.al., 2014). In spite of the lack of consensus, there seems to be an increasing trend toward teaching the fundamental concepts of computer science, while learning a programming language is to be used as a tool to assist with this process. Therefore, the goal is not for the students to master a specific programing language, but rather to learn the concepts, develop the necessary skills and become familiar with technology so they can be best prepared for a future heavily based on technology.

What are students gaining from learning computer science? According to Mitch Resnick (TED, 2013), director of Lifelong Kindergarten group at the Media Laboratory at Massachusetts Institute of Technology, the most important concept computer sciences teaches children is the core principles of design:

> How to start with a gem of an idea and turn it into a fully-fledged functioning project.". This process involves "experiment with new ideas, […] take complex ideas and break them down into simplier parts, […] collaborate with other people on your project, […] find and fix bugs when things go wrong, […] keep persistent and persevere in the face of frustration when things are not working well (TED, 2013, 13:40).

These skills are not only important for professional life but they have applications in personal life as well. Many others (Solomon & Rusev, 2008; Heese, 2014; Sterling, 2015; TED Partovi, 2014; TED Waterhouse, 2015, Crow, 2014) align with Resnick's beliefs.

Computer science promotes computational thinking, which trains students to think in logical and analytical ways. Computer science students become familiar with algorithms and develop skills that allow them to break down a complex problem into smaller manageable parts, and then to create a sequence of steps that will lead to the solution. With programming, students can enhance their understanding in other curriculum areas, like mathematics. For example, they can gain a better understanding of variables and appreciate their importance, or they can recognize that the long division they have been struggling with is just another algorithm. They can understand how to generate random numbers when they need to test their hypothesis about various probabilities (Solomon & Rusev, 2008). With computer science, students hopefully advance their problem solving skills and learn to generate multiple solutions to approach the same problem. Due to the interactive environment of programming, students can test their solutions and self-identify what went wrong and think of new possible solutions to their problem. In order to succeed they need to develop persistence and perseverance, both important values for every aspect of their lives.

Not all problems will have successful or have the most efficient solutions, so students will need to reach out for help. Peer work and collaboration are two other important skills students can develop while working in computer science projects. Much of programming is accomplished in teams where students can communicate, test their own ideas, build on the ideas of others, ask questions and work together towards the solution. Programming is an area that can bring students with different skills and interests together. For example, in an assignment about designing an app, the teacher could have each student in a group focusing on a different task. While some will do the coding and math, others will have an interest in the design or in testing the app with users and provide feedback to the team.

Working in groups, validating ideas and applying peer feedback, is a process that allows students to embrace their mistakes and view them as part of their learning process. Computer science is one of the few subjects that allows students to do this. The notion of "debugging" is at the core of programming, so students must realize that learning is about fixing and learning from mistakes, rather than not making mistakes at all. Students' perceptions toward "failure" change since they can just return to their computer program, examine it and make changes to solve a problem (Mauch, 2001). Once students are released from the fear of making mistakes, good things can happen. Students will be more willing to try new ideas, to play and experiment, to take risks, and be creative. Once reaching this stage, together with building confidence with technology through learning computer science, students can then express themselves with technology. They can create technology that is meaningful for them and aligns with their own needs and interests. A good example of this learning process is the one of Callum Pickles (CBC, 2014), a 15-year-old boy who designed a computer application to help prevent cyberbullying in his school community.

Early exposure to computer science and coding in schools will assist students realizing the value of this educational discipline and the variety of careers they can follow in this field. In March 2016, the Information and Communication Technology Council (ICTC) released a report revealing that by year 2019 there will be a demand for over 180,000 skilled ICT professionals in Canada. The report notes that unfortunately the domestic supply of ICT graduates and professionals will be unable to meet this demand. South of Canada, the U.S. Bureau of Labor Statistics (BLS) estimated 1.3 million job openings in computer and mathematical professions by 2022 (Gallup, 2015). In Europe 825,000 ICT job vacancies are estimated by 2020 (European Commission, 2014, par.3). This job forecast is not only about the jobs in the ICT field. Every field, from medical and energy to entertainment and transportation, will be affected; every field will require people literate in computer science and programming skills (TED-Partovi, 2014). The ACM/IEEE curricula for computer scientists acknowledges the need for computer science in all fields, and is guided by the principle that it "should provide students with the flexibility to work across many disciplines" and "should prepare graduates for a variety of professions" (ACM/IEEE, 2013, p.20). By not giving students the opportunity to learn and develop those skills at school, teachers will not be preparing them for a large range of future career opportunities.

On the other hand, critics (Trucano, 2014; Marx, 2016; Farag, B.) say that it is unknown what the jobs of tomorrow will be, and so, deciding what students should learn at schools should not be based on what job-relevant skills the job market requires right now. Advocating initiatives for coding are often driven or supported by the large high-tech companies. Many are suspicious about their motives. Dash (2016) argues that our schools should not become a pipeline feeder of employees for the high-tech sector. Education curricular policies should be influenced by pedagogical value rather than future job opportunities. Even though the importance of acquiring skills such as problem solving, logic and analytical skills is not questioned, critics argue that computer science and coding are not the only opportunities students have to develop those skills (Trucano, 2014). In addition, education policies need to ensure student competency in the basic literacies of reading, writing, and arithmetic is in place, before adding new ones to already full curricula.

Promoting coding as a ticket to economic prosperity and future job security for the masses is perceived as dishonest (Farag, 2016). Similarly, some argue that advocating for "Coding as a new literacy" is deceitful. Marx (2016) notes, "It's undeniable that coding is a hyper-important skill in the 21st century — but it's not the end-all, be-all of literacy. Literacy spans a variety of languages, communication tools, and colloquial, idiomatic trends. There is no "one" magic bullet." (para.11). Coding and computer science is not for everyone. Not everyone will like it, be good at it, or be willing to pursuit it as a career in life. Making it sound like people who cannot code will be the illiterates of the 21st century raises ethical issues.

Even if there was a universal consensus that all children should learn computer science, implementing this idea faces major barriers. According to a recent survey in US schools conducted by Gallup (2015 & 2016) on behalf of Google, the majority of principals and superintendents reported that there are insufficient funds to train or hire computer science teachers, and too few teachers have the necessary skills and qualifications to teach the subject in schools. In addition, since computer science is not part of the mandatory curriculum, schools have to devote most educational time and resources to mandatory courses instead. Many schools reported not receiving a high level of demand for computer science education among students and parents in their communities, and this holds especially in rural communities. The lack of equipment and software were also mentioned as barriers to offering computer science in schools.

When computers were first introduced in schools the focus was on learning programming and how computers work (Heggart, 2014). Then, and since graphical user interface software applications were in its infancy, the user had no other option other than to learn to program the computer and make it do the work she wanted to do. Once software applications became interactive, user friendly and widely available, there was a shift to start using computers only as tools. This has resulted in having mainly users of technology rather than creators of technology. Thus, in my opinion, this "coding movement" is about returning back to basics: understanding how computers work, their power and limitations.
Having students understand how computers work helps them understand the world around them, which is constantly becoming more and more high-tech based. With advanced technology, automation will significantly progress and machines will be taking over more manual labour jobs in the future, leading to an even more knowledge based economy. Investing in technology professions might be a way to economic development and prosperity for our communities. Initiatives in the Eastern Canada are viewing opportunities with technology as a way to retain their youth in the Atlantic Canada communities, who might otherwise migrate West due to the scarcity of traditional jobs in the home area. Countries like Estonia have invested in technology and succeeded (CBC, 2014). Ireland, a country caught up in the 2008 financial crisis, has developed a very throughout action plan with the goal to "Make Ireland a Global Leader in ICT talent" (Government of Ireland, 2014). Pupils with increased exposure to computer technology are more confident in their own skills and more likely to consider learning computer science in the future (Gallup, 2015). Not exposing them to this field is like not giving them the necessary tools to meet the opportunities of the future.

Exposing students to computer science does not mean that every student will end up being a computer scientist or pursuing a career in the high-tech industry. Similarly, students are currently exposed to math and arts with no expectations of them all becoming mathematicians and artists. Students should graduate from school with a broad range of knowledge, skills and experiences in all disciplines; they are to be prepared to pursue their own passions and interests, regardless of what jobs will be available in the economy at the time of their graduation. We have to acknowledge though that "Computer Science impacts nearly every modern endeavor "(ACM/IEEE, 2013, p.20), therefore regardless of the profession students are going pursue, computers will likely be dominating their professional and personal lives. In the 21st century, computer science should not be ignored from this broader knowledge and skill-set we want our students to have in order to be successful in their future lives.

There has never been a better opportunity for this to happen. Currently, there is a wide range of available online resources (e.g. Code Academy, Khan Academy, Scratch, Tynker). Most of them are available for free, and many of them provide detailed lesson plans, books and other supports for both students and teachers. In fact, the challenge for the teacher is that there is a plethora of resources available and more are produced each year, therefore it makes it difficult for the teacher to decide what are the best ones to use to teach computer science. No teacher without extensive research or official guidelines will be confident in a selection of resources to use, and no teacher will have the time to try them all and make this decision. For this opportunity not to be missed, the teachers need to have some guidance from the Ministries of Education. The resources are online and available for free, but there should be some recommendations from specialists in the computer science education field on which ones should teachers be using. Coding Resources Connected to BC's Curriculum is a good example in this direction.

Providing every teacher with these supports, the teacher can then take the role of a facilitator in class, while students can become self-directed learners, moving at their own pace, collaborating with their classmates, engaging in further research and finding solutions to their problems. It is these sorts of skills that we want our children to gain and utilize, so they can become life-long learners and meet the demands of the technology oriented 21st century.

References:

ACM/IEEE Computing Curricula Task Force. (2013). Computer science curricula 2013: Curriculum guidelines for undergraduate degree programs in computer science. *Association for Computing Machinery (ACM) IEEE Computer Society.* Retrieved from: https://www.acm.org/education/CS2013-final-report.pdf

Carroll, D. (2015). Should we really teach all kids to code? Retrieved from https://www.linkedin.com/pulse/teach-all-kids-code-why-how-david-carroll

CBC Documentary. (2014, December 4). Code Kids. Retrieved from https://www.youtube.com/watch?v=6JGy8zmskbM

Crow, D. (2014). Why every child should learn to code. Retrieved from https://www.theguardian.com/technology/2014/feb/07/year-of-code-dan-crow-songkick

Dash, A. (2016). It is more than just "teach kids to code". Retrieved from https://medium.com/humane-tech/its-more-than-just-teach-kids-to-code-177fd6cb7184#.3z9jtq8vs

Ellis, L. & Corcos, S. (2015). Is It "Computer Science" or "Programming"? Retrieved from https://medium.freecodecamp.com/is-it-computer-science-or-programming-c01383dacc9c#.hu5bha8z1

European Commission. (2014). Coding: The 21st Century Skill. Retrieved from: https://ec.europa.eu/digital-single-market/en/coding-21st-century-skill

Farag, B. (2016). Please don't learn to code. Retrieved from https://techcrunch.com/2016/05/10/please-dont-learn-to-code/?utm_content=buffer0644b&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer

Franklin, D. (2015). Putting the computer science in computing education research. *Communications of the ACM*, *58*(2), 34-36.

Frelix, E. (2016). Coding and computer science should be mandatory in Canadian schools. Retrieved from http://www.theglobeandmail.com/report-on-business/rob-commentary/coding-and-computer-science-should-be-mandatory-in-canadian-schools/article31456908/

Gallup (2016). Trends in the state of Computer Science in U.S. K-12 schools. Retrieved from http://services.google.com/fh/files/misc/trends-in-the-state-of-computer-science-report.pdf

Gallup (2015). Searching for Computer Science: Access and Barriers in U.S. K-12 Education. Retrieved from https://services.google.com/fh/files/misc/searching-for-computer-science_report.pdf

Government of Ireland. (2014). ICT Skills Action Plan. Retrieved from: http://cork.etb.ie/wp-content/uploads/sites/20/2014/08/14042014-ICT_Skills_Action_Plan-Publication.pdf

Heese, B.(2014). Why Teach Computer Science in High Schools?. Retrieved from http://www.huffingtonpost.com/brian-heese/why-teach-computer-science_b_4350315.html

Heggart, K. (2014) Coded for success: The Benefits of learning how to Program Retrieved from https://www.edutopia.org/discussion/coded-success-benefits-learning-program

Information and Communication Technology Council (ICTC) (2016). Digital Talent: Road to 2020 and Beyond. Retrieved from http://www.ictc-ctic.ca/wp-content/uploads/2016/03/ICTC_DigitalTalent2020_ENGLISH_FINAL_March2016.pdf

Lee, S. (n.d.). What's the Big Deal about Coding? Retriecved from http://www.figur8.net/whats-the-big-deal-about-coding/

Marx, A. (2016). Stop telling me I need to code. Retrieved from: https://medium.com/startup-grind/stop-telling-me-i-need-to-code-7ab67681304c#.p0asy3mqw

Mauch, E. (2001). Using technological innovation to improve the problem-solving skills of middle school students: Educators' experiences with the LEGO mind storms robotic invention system. *The Clearing House*, *74*(4), 211-213.

Obama, B. (2013, December 8). *President Obama asks America to learn computer science*.

[Video file]. Retrieved from https://youtu.be/6XvmhE1J9PY
Solomon, J. & Rusev, P. (2008). Early Acquisition of Computer Science. Retrieved from: https://cs.stanford.edu/people/eroberts/cs181/projects/early-acquisition-of-cs/advantages.html
Sterling, L. (2015). An education for the 21st century means teaching coding in schools. Retrieved from https://theconversation.com/an-education-for-the-21st-century-means-teaching-coding-in-schools-42046
TED. (2013, January 29). Mitch Resnick: Let's teach kids to code [Video file]. Retrieved from https://www.youtube.com/watch?v=Ok6LbV6bqaE
TED. (2014, December 8). Computer science is for everyone | Hadi Partovi |. [Video file]. Retrieved from https://youtu.be/FpMNs7H24X0
TED. (2015, February 10). Teach kids code | Torgeir Waterhouse | TEDxOslo. [Video file]. Retrieved from https://www.youtube.com/watch?v=-TP7Zjqmj_4
Trucano,M.(2014). Should all students learn how to code? Pros and Cons. Retrieved from http://www.wise-qatar.org/coding-cognitive-abilities-michael-trucano
Trudeau, J. (2016, January 14). *Prime Minister Justin Trudeau talks importance of coding, algorithms*. [Video file]. Retrieved from https://globalnews.ca/video/2453986/prime-minister-justin-trudeau-talks-importance-of-coding-algorithms
Wilson, C. (2017). Code-ifying the Curriculum: Bringing computational thinking into the classroom. *Learn: The magazine of BC Education.* Retrieved from *http://www.bcteacherregulation.ca/documents/Learn/2017/LearnMagazine_Spring_2017.pdf*
Yadav, A.,Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Trans. Comput. Educ. 14, 1, Article 5*. DOI:http://dx.doi.org/10.1145/2576872

Eleni Galatsanou Tellidis completed her undergraduate degree in mathematics and computer science, and master's in financial mathematics in Europe (Greece and UK). She has worked as a researcher and project lead for Brandon University's Rural Development Institute since 2013, and has nine years of working experience in the financial services industry both in the UK and in Greece. With a goal to become an inspiring math and computer science teacher she is currently working on the Bachelor's of education after degree program at Brandon University.

MSc, B.Ed. 2nd year student
email: eleni.gal33@gmail.com
Twitter: @elgal

Chapter 12 - It's Not Another Math Course: Teaching Prospective Teachers A Way to Teach Computer Science

Dr. Reynold Redekopp
University of Manitoba

Wake-up call #1

A number of years ago I was teaching math and computer science at River East Collegiate. After our first period grade 11 math class I was walking down the hall to my computer science class with a student I taught in both subjects. After a bit of an epiphany, I turned to her and said, "This is really just like going to another math class isn't it?" She agreed.

And that changed everything for me.

I began to change how I taught CS, especially in the kinds of assignments I gave and examples I used. I tried to find non-math or more visual examples and revised assignments so that the program could also include some creativity - a visual/audio feature or output. I tried to make it 'not another math course' so that it would still appeal to those who had issues with math. The math is unavoidable, but the way you get there can be way more interesting.

Wake-up call #2

There is a group of CS teachers that gets together some SAGE PD days (a province wide Manitoba PD day where you can meet with teachers of a similar subject). I asked a CS prof at the University of Manitoba how many first year students in CS had never taken CS in high school. The estimate at the time was about 70%, now he thinks it's about 50-50. The prof also said that they were generally as successful as those who had taken high school CS. Finally, he noted the current average of women in the CS Faculty is 13.5% but Honours students in the Co-op program is up to 24%.

That changed things some more.

If high school CS was not needed for first year CS entrance or success, then there was a lot more flexibility in what I could do with my students and in what I could ask them to do. I would not have to follow all those CS textbooks that were essentially a first year CS course for high school students. I could create projects that were more interesting for the non-math types. We could be far more creative solving problems, and try projects that were still challenging but that deviated from the 'text' curricula. These projects often pushed us further than a student would have gone because they really wanted to get it done. This also occurred in my multimedia courses where we were doing a lot of scripting, but there were even more options for creative design and output.

Wake-up call #3

I have visited many schools and talked to many CS teachers. It seems the norm that few students remain in CS from grade 10 - 12. I taught in a large high school where we had 3 sections of CS in grade 10 and usually ended up with one section by grade 12. Some years we had to put the grade 12 students in with the grade 11 group so we could offer the credit. Something needs to change.

Mindset changes are necessary for this shift to happen. We need to attract and keep a wide variety of people in CS: women and other underrepresented groups in particular. This will improve our IT preparedness as a community, but involving these underrepresented groups may also bring about a shift in our thinking about CS and take us in new directions of computational thinking. This could be critical as we confront some of the enormous problems we face, and will face, in the ever more connected community. We need more variety of thinking among those who will solve these problems.

So what to do?

I now teach at the Faculty of Education of the University of Manitoba and teach the Computer Science Methods course. With the above ideas in mind I have really had to think about how I influence prospective teachers about teaching computer science. Most of my students have seen only one way to teach CS - a very traditional way. Thus it is hard for them to imagine alternative ways of approaching the subject. Fortunately they are generally willing to think about the big ideas in CS and how we might approach them differently without sacrificing any integrity.

We begin by looking at the big picture and the big concepts and realize that learning code is only a part of what we should be teaching. The Manitoba Education CS Framework is clear about this. The framework has four General Learning Outcomes (GLOs):

- Human Relations,
- Literacy and Communication,
- Problem Solving, and
- Technology.

It is only in the Technology section that programming is mentioned, but this tends to preoccupy many high school CS courses while the problem solving component tends to be based on math related problems.

Within each GLO there are a number of Specific Learning Outcomes (SLOs) such as Society and the Environment, Ethical Behaviour, and different facets of communication. There are lots of interesting possibilities there for teachers who choose to work on this part of the Framework. The part I work on most is how to approach the computational thinking and programming agenda with lesson of an emphasis on math examples and problems.

Don't let the programming language get in the way of the idea.
The main ideas in Technology section of the Framework are:

- Programming standards (eg. naming of variables and subprograms)
- Data structures and types
- Control structures
- Debugging techniques
- Reusable code (libraries)
- Subprograms and parameters
- Objects
- Algorithms

Languages such as C++ and Java have a lot of overhead that a student has to encounter before they get to the main (yes, that might be a pun) idea. There are many other programming languages that allow the user to play with the idea in an easy-to-use interface. Here are some examples that might work better for some students. These may help them understand the idea before they try to implement it in a high-end language. A few examples follow.
NOTE: Any of the examples need only take a class or two to get the concept across before students try it in a more formal language, but after doing so the concept is much more accessible.

Using Logo to understand loops and objects.
Logo was developed in the 1970's by Seymour Papert at MIT. You may know it as 'turtle graphics.' The language has no overhead at all.

One of the most common lessons in Logo is how to draw a square: **Repeat 4 [fd 100 rt 90]**. This a simple but powerful way of understanding loops. However the square can easily become a procedure:

```
To Square
Repeat 4 [fd 100 rt 90]
End
```

Now if you want to draw a square you simply type **Square**. You can also move on to parameters to draw a square of any size you pass a parameter :size to the procedure

```
To Square :size
  repeat 4 [fd :size rt 90]
End
  Square 50
```

This will create a square with sides of 50. Square 125 will have sides of length 125.

Finally you can **easily demonstrate recursion**. In this example the procedure **tree** calls itself until the end condition is met. With Logo you see the recursion being drawn on the screen.

```
to randomcolor
  setcolor pick [ red orange yellow green blue violet ]
end

to tree :size
  if :size < 5 [randomcolor forward :size back :size stop]
   forward :size/3
   left 30 tree :size*2/3 right 30
   forward :size/6
   right 25 tree :size/2 left 25
   forward :size/3
   right 25 tree :size/2 left 25
   forward :size/6
   back :size
end

clearscreen
tree 150
```

Debugging is a natural part of programming. With visual programs like Logo, Kodu, Scratch, GameMaker, etc., some of the debugging techniques make more sense and are more immediately obvious.

Programming Concepts with Scratch

Scratch is a very popular programming language designed for younger students. It is all drag-and-drop programming with different kinds of commands (motion, events, looks, etc) shown in different colours. The 'OurScratchProject' wiki explains it like this:

> Scratch supports these concepts: sequence, iteration (looping), conditional statements, variables, threads (parallel execution), synchronisation, real-time interaction, boolean logic, random numbers, event handling and user interface design.

There is more too. One of my former students read this chapter and corrected me, saying:

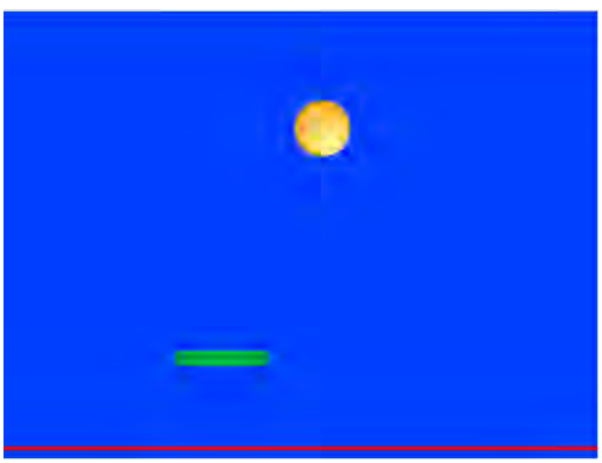
> I would want the focus more on implementing Scratch over the other programs ... it is almost robust enough to be an entire semester worth of material if educators are creative

> enough with bigger projects. It can REALLY do a lot. It has a great hook, a really well done UI that is intuitive to navigate, and has become extremely powerful. I had my students build some impressive programs with it, even slipped in some math when they weren't looking. It is absolutely and unequivocally the best tool available to teach almost any variety of intro CS. (Roque Lacroix)

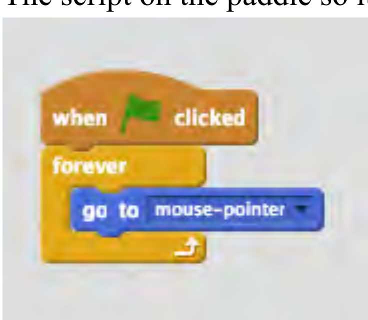
All of this is done in a visual environment, so like Logo you can see what your code produces as it happens. Admittedly, with the speed of modern devices this can all seem simultaneous. A surprise to me was when one of my student teachers came back and reported that he had to explain to students that a program ran in sequence, not 'all at once.' Students had become so used to things 'just happening' on their screens that they didn't understand the sequential nature of a program. Scratch and similar visual programs can help with that, especially if you include timers to slow down the actions.

Again, it's important to note that all of these sample programs I am demonstrating here have limited use in terms of 'real-world' programming but in **a day or two** can be really powerful in terms of illustrating difficult concepts in computational thinking.

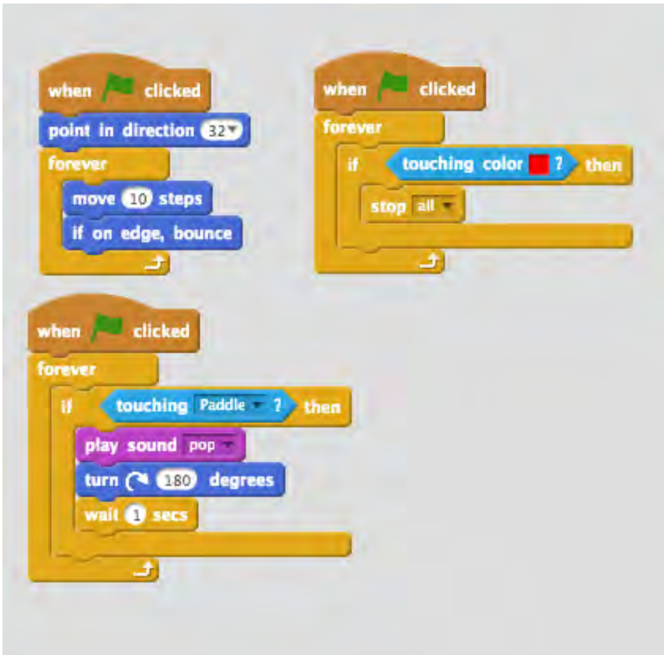
Here's a quick example from the tutorial in Scratch: a simple game to keep the ball from hitting the bottom:



The script on the paddle so it follows the mouse:



The script on the ball that keeps it moving, bounces off the paddle and ends the game if the red line is hit:



Hopefully you can see the various concepts that can be drawn out from this: sequence, iteration (looping), conditional statements, threads (parallel execution), real-time interaction, and event handling

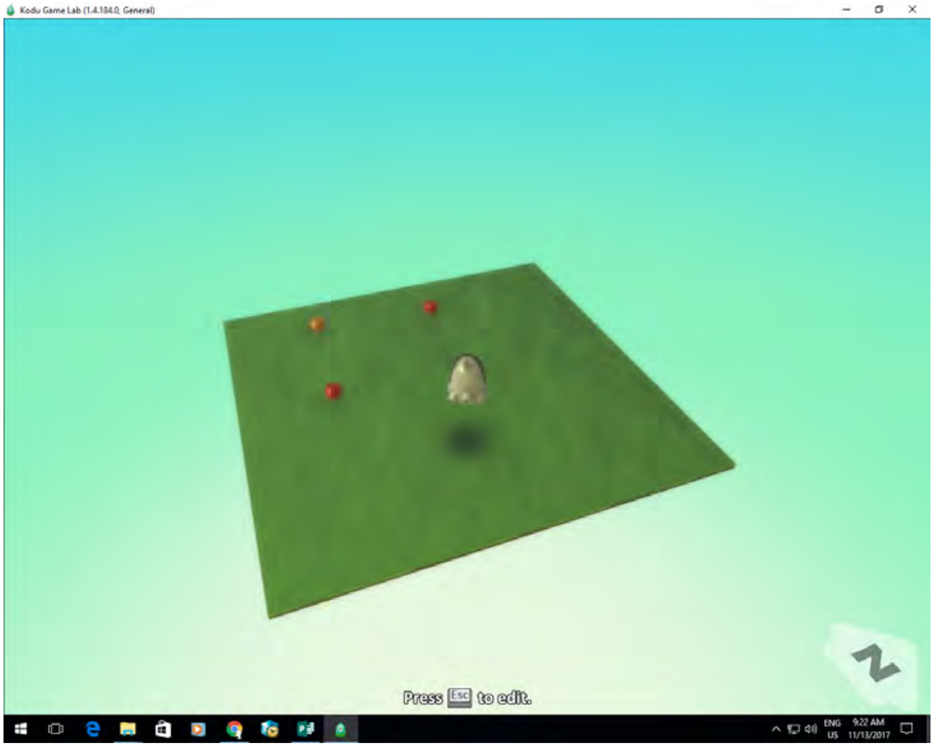
Kodu - Objects and Classes
One more example from among the many easy to use options. Again, this need only take a day or two or a part of different classes to introduce the concept.

Kodu is a 3D programmable game world which is accessible for students from Grades 2 - 12 (perhaps even grade 1 but I have definitely seen grade 2 students create games).

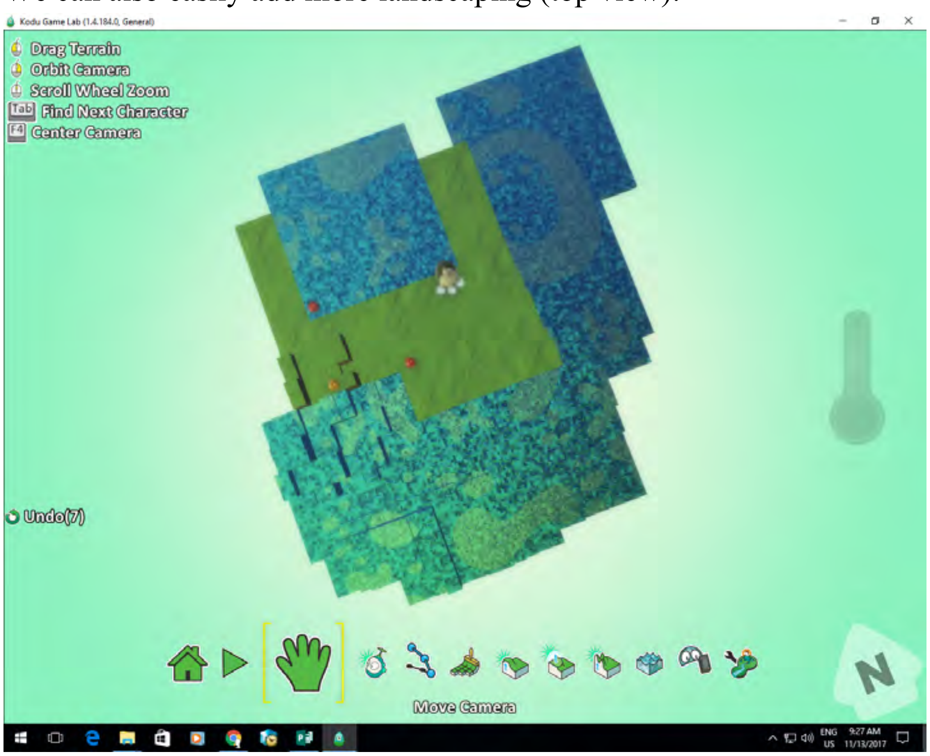
Besides being a rich 3D world, Kodu has a unique programming/logic system of When/Do, which is a terrific way of introducing students to conditions and actions.

Kodu allows students to create a 3D world by adding landscaping to the playing area, They then create objects and add characteristics to them as well as program them. The objects can interact with each other, scores can be kept and the game can have a definite end. In the very simple example that follows, you will see objects and classes, sequencing, random numbers, variables, interaction, conditionals, and event handling. You can tease out the meaning within any or all of these for your students.

First the setup. In this scene we have a Kodu object, two red apples, and an orange apple; each has been placed by clicking on the playing field and choosing from a variety of objects.

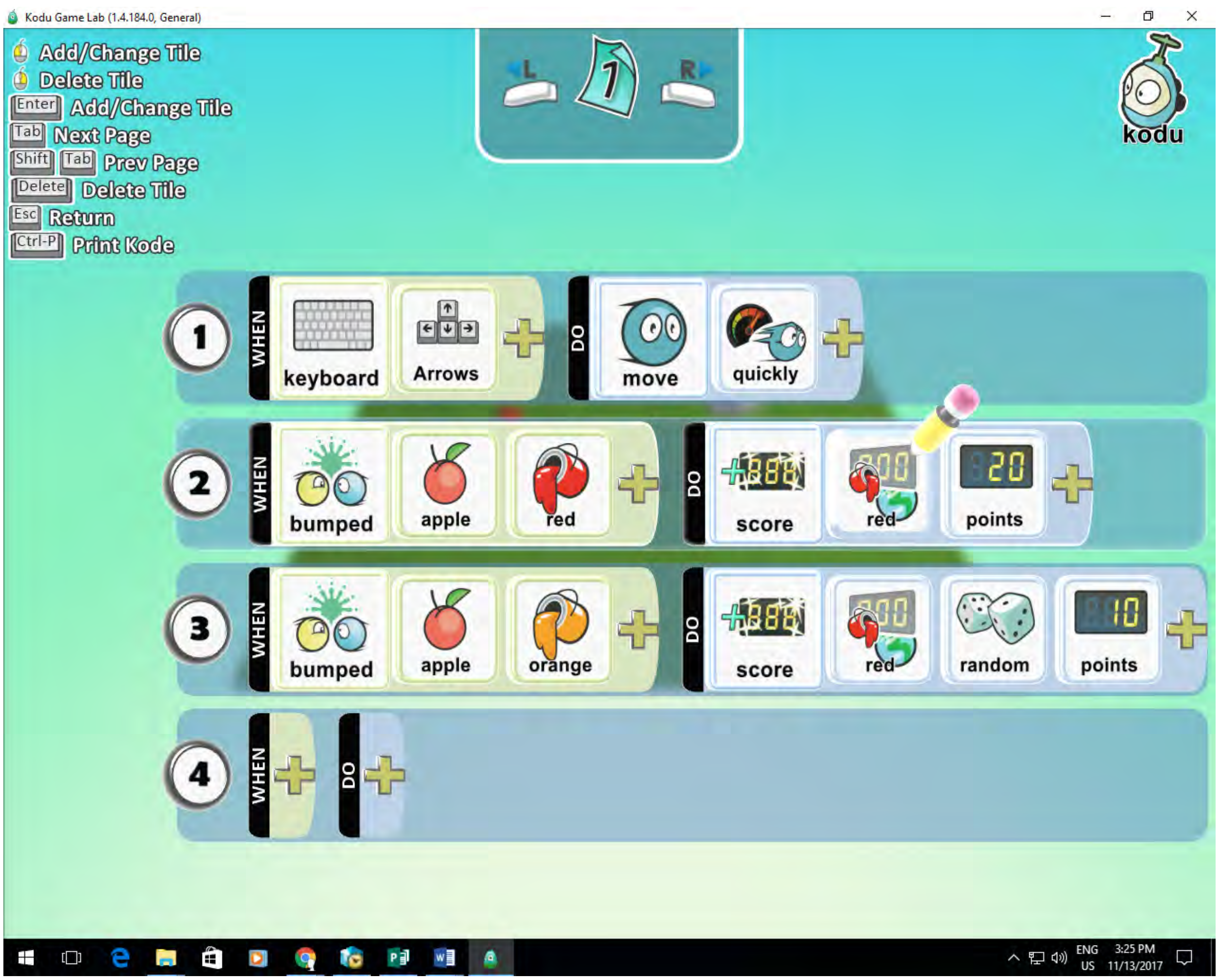


We can also easily add more landscaping (top view):



But my favourite part of Kodu is the programming logic that is built in. It is a “When / Do” logic. First, students choose (by right-clicking) the program option on any object. Then they are offered the When/Do choices. In the following screenshot of the ‘coding’ on the Kodu character we see:

1. **When**: the arrow keys are pressed; **Do**; move quickly
2. **When**: Kodu bumps into a red apple; **Do**: add 20 points to the Red score
3. **When:** Kodu bumps into an orange apple; **Do**: subtract a random number of points (up to 10) from the Red score.



All this is done by clicking and choosing from among the options. No keyboard entry is required. So rather than worrying about missing semicolons or variable type mismatches, student can focus on the logic and debugging process. To have students get to this point in Kodu takes about 15 minutes.

What's the takeaway?
There are some fun ways to have students learn big ideas in computational thinking and coding. That said though, it is still incumbent on the teacher to draw the connections between these big ideas and what their students are doing in the 'regular' part of the course. We can't assume that students will make the transfer on their own. Teachers need to make the connection explicit and directly related to the rest of the program.

What is intuitive in Kodu or Scratch or Logo is still not necessarily intuitive or easily transferred to Java or VB or C++ or any other high level language. It needs to be connected for most students.

What if I don't like Scratch, Logo or Kodu?
The programs I am suggesting here are no substitute for a high-level language. You run into limitations fairly quickly if you want to develop more sophisticated programs. There are other great possibilities like GameMaker, Alice, GameStar Mechanic and others. All of them are good for some things and have their limitations which show up fairly quickly, but they are all very good at illustrating certain computational thinking ideas and should be used as such.

What about students who still struggle with the big ideas in a higher level program?
Let them continue in Scratch or whatever platform you have decided to use. They will likely develop projects that will surprise and amaze you - or not. If you decide on the criteria you want to see in a project, they may not meet all of them but are more likely to achieve some success and satisfaction than if they struggle with your chosen language.

Does this weaken the integrity of your program? No. It merely gives more students an understanding of CS and an opportunity to grow in the area. This may lead them to continue in CS and being able to move into a higher level language at some point, or they may just leave CS, but they will do so with a better understanding and appreciation to take into whatever else they choose to do.

Are these the only options?
I have given some examples of ways to help students develop some concept of the big ideas. There are many other options. Robotics, drones, and animation are other examples that provide access to computational thinking ideas. They often have programmable aspects built in and so they can be an exciting way for students to approach programming.

Other examples from some of my CS Methods students:

Using RPGmaker to Teach Computer Science - Brent Robertson
Arrays in Alice - Scott Hardman

Recursion another way - Alex Kozub
Teaching Boolean Logic, Variables, and Objects Using Kodu - John Reimer (note: the use of the word 'funner' is my fault, not John's. It's an in-class joke)
Ideas for Teaching Sorting and Algorithms - Samantha Adamson
Teaching Boolean Logic with Kodu - Mitchell Wiebe
Teaching Arrays Using Scratch - Jon Jonasson

Conclusion

This chapter does not pretend to provide an exhaustive list of ways to introduce difficult CS ideas to students, but Instead, simply start you with some ideas and directions you might take. What you can actually implement depends on your computers and software, your background, your students' interests, and your teaching preferences.

This is certainly not a call that all students should become computer scientists. It is my hope that by expanding our repertoire of concept development tools we can encourage more of our students, particularly women and other underrepresented groups to consider taking courses in technical fields like CS so that they have more opportunities, and so that our digital world creator population becomes more diverse.

Reynold Redekopp has been using and thinking about digital technology since someone brought a Commodore Pet into his classroom and asked him to see if there was any educational value to these things: despite no connectivity, no resources, and a cassette tape player to save your program. At least it had a whopping 32 Kb of RAM!! That has led to a lot of experimentation, looking at how to use each generation of computer/device in effective ways.

Chapter 13 - Student-Centered Learning in Computer Science: Allowing Students to Choose Their Own Curriculum

By Matteo Di Muro

Vincent Massey High School (Brandon), Grades 10-12
Email: matteo_martello@hotmail.com

With the rise of technology availability, many schools are striving to create 1-1 technology educational environments with authentic tasks (Edwards, 2013; Herrington, Reeves & Oliver, 2014). With the 1-1 educational technology in a classroom, students can engage in self-paced learning, allowing for the teacher to offer the optimal support needed by students at all learning levels (Edwards, 2013). It was with these ideas in mind that I one day found myself in my principal's office, inquiring about whether he would consider reviving the long dead computer science program our school once offered. He agreed it would be worth a shot, and five years later, numbers in the program have grown from one course offering per school year, to six course offerings per school year, with courses ranging in level from grade 10 to grade 12.

With this growth in computer science offerings, I had a unique opportunity to attempt to implement a student-centered inquiry model of instruction and assessment in a computer science high school classroom. Being the only teacher in charge, and having a supportive administration team on my side, I went ahead and tried a variety of different learning models in my own classroom, until I eventually settled on a student-centered "inquiry" approach as being the most conducive to learning in this sort of environment. The basic tenant of inquiry learning is that students understand a subject by engaging in self-directed investigations.
I put quotations around inquiry because my method does not actually satisfy the definition of true inquiry—however, I feel I have struck a balance with scaffolding and true inquiry that works in the context of this subject.

I generally do not rank learning philosophies in the terms of "which is best"; rather, they all have their own unique strengths and weaknesses—the effective teacher learns how to navigate among them, and evaluate which methods are best suited to the student and to the learning goals at hand. My belief is that there is no such thing as a "best practice" in teaching--merely teaching practices that are more suited to certain goals and learning contexts. For this reason I would never say one teaching approach and atmosphere should always be used--they must be used in a context that plays to their strengths and minimizes their weaknesses. In providing my students with an authentic inquiry learning environment, it is important to understand that while I think the best way to describe my overall approach is student-centered with dashes of inquiry along the way, this may not always be the case all the time in the course! In fact, I make use of direct instruction, peer instruction, online instruction, guided inquiry, microteaching, and a lot of scaffolding, especially in the grade 10 and 11 course. Using a mixture of methodologies and techniques allows me to play to the strengths of each technique as required by the context of the learning goals at hand, while leaving the weaknesses behind.

The purpose of this reflective article is to examine and discuss the benefits and drawbacks to implementing a student-centered approach to learning computer science in a high school setting. I will discuss the techniques used, the underlying philosophies of the teaching methodology supporting the instructional techniques, and the resulting successes and challenges encountered in the typical day to day instruction of the course in this manner. Themes to be investigated are how ~~does~~ the power structure of the classroom shifts and changes between teachers and students; how time is managed within the classroom; how assessment of student products is changed; and how students perceive themselves as learners, and how they perceive their teachers. Also investigated are the roadblocks to success, the ways I have overcome them, and the perceived benefits of this program.

Setting

Students range from grade 10 to 12, thus their ages are 15 to 18 years of age in a high school setting. Class sizes range from 12 students up to 29 students. Computer science is a non-compulsory credit; thus, most students who take these courses are usually already interested in the subject of computer science for one reason or another. The academic backgrounds of these students are varied. Generally, research has found that math and English grades are predictive of success in most learners at this age; this has not necessarily been my experience in teaching in a student-centered classroom.

Teaching Techniques: What does class look like on a typical day? Taking it one course at a time.

It is important to note that not only are there specific aspects to how I teach at each grade level, but that I have designed the three levels of the course, ranging from grade 10 to grade 12, with the idea in mind that many students require scaffolding and some encouragement to feel and be capable of succeeding in a student-centered approach to learning. The end goal is that by the time a student begins the grade 12 course, they are fully able to choose what they wish to learn, how they plan to learn it, and begin to seek the resources required for their learning as they need them, with minimal input from the teacher. I shoot for a breakdown that looks likes this:

Table 1. A comparison of the amount of material chosen by the teacher, versus how much "material" is scaffolded but chosen by the student.

Course	Prescribed Course Material/Student Chosen Material
Grade 10	80/20
Grade 11	50/50

Grade 12	10/90 (this may also be closer to 0/100 depending on the student)

Grade 10 Course

The objective of the grade 10 course is two-fold and key to the success of the grade 11 and 12 courses: 1) to learn how to code using Microsoft Visual Basic, a user-friendly language, and 2) to teach students how to be effective self-directed learners in computer science.

When I started planning this course, I was just at that point in my teaching career, as a third year teacher, where I was beginning to examine research literature. I was considering a master's program, and couldn't help myself. I considered myself a constructivist, and I knew I wanted to take a student-centered approach with some inquiry tasks. In current inquiry literature, there is a debate as to the amount of structure that should be provided to students. Lazonder (2014) found that "unguided discovery learning is less effective than direct instruction" (p. 454), whereas Zion and Mendelovici (2012) argue that structured inquiry is but "one pillar of the scientific inquiry process … therefore, structured inquiry works well only for developing basic inquiry skills" (p. 384). The need for a more open approach to inquiry is clear from these two articles. Recognizing the reasoning behind both arguments, I decided that the grade 10 course must slowly introduce student options and inquiry, with much scaffolding and feedback from teacher to student and student to teacher. In grade 10, I have made most of the decisions as to *what* students will learn, but have modified the time frames in which they learn it, and how they learn it. I have removed myself as the primary delivery mechanism of what they are to learn. This builds up to the "exam" in the course, which is a project that is chosen by each individual student. I also provided "pre-canned" projects that are either from other resources or that I have designed; as some students are not ready to choose their own projects by the end of the course. This sort of approach to inquiry is supported by McIntosh (2012), who explains how you can guide inquiry for learners by calculating what resources you provide for them in the learning environment.

I decided to create my own guided notes. These make use of online resources which begin to show students how they can use the internet to find credible learning resources, and add to the structure of projects and how they are assessed. They are designed to slowly wean students off the need for their teacher, especially in the sense of the teacher choosing how students display their learning, and what they will learn. Even when a student chooses to create their own project (we do about 3 major ones), I support them through the use of a planning process (Appendix A)—remember, scaffold, scaffold, scaffold!—where they must pitch their idea to me, answering key questions that will help them plan, and help me as their teacher ensure that I can support them with the required resources.

By a few weeks into the course, a student is more likely to ask a peer for help than to ask me (although I am generally busy helping students either way with problems peers could not solve).

While some students may initially thrive in an unstructured, student-centered model right off the hop, the vast majority of students, in my experience, need to learn skills in becoming self-sufficient learners who no longer dependent on teachers. This process can take weeks to months before a student is capable of being independent 80% of the time. And I will not lie, some students are definitely more ready to do this than others, although you can help any student improve greatly. It takes time to move from pure direct instruction (which is the teaching style most students, including university students, are familiar with) to a model where the student is responsible for their learning and the teacher is responsible for ensuring that effective resources and routines are in place to ensure student learning takes time. While effective for this particular course, it is key to uproot the idea that "the teacher tells me what to do and lectures me for most of the class on how to do it" to "I know how to be an independent and effective learner on my own, who can use resources provided for me, and the teacher is another resource I can depend on" is one of the major points of the grade 10 course.

One last dire warning! Students will experience more cognitive dissonance with this sort of proposed structure, and it is imperative that as a teacher you ensure this dissonance is not too overwhelming or lasts too long. Basically you have to ensure that students can get help in a timely manner, that nothing is hard for the sake of being hard, and that guided notes, exercises, tasks, and expectations are clear. If something is going off the rails, it is a good time to perhaps teach a quick 5 or 10 minute lesson on the topic that is giving students grief. Whether it is a small group of students or the entire class--it is important to recognize when this is the case. I frequently will ask my class for impromptu surveys of whether or a topic is becoming assimilated or whether more direct guidance is needed as a class. This is especially key during the first 4 weeks of the course; you want to ensure students experience success--not that they throw their hands up in frustration and decide computer science is not for them.

Structure of the Grade 10 Course.

To this end, I have created electronic notes for every topic covered in the grade 10 course. (for a sample course outline, including a list of topics, see Appendix B. Appendix C and D are for the grade 11 and 12 courses). Students are directed that a package of notes will take x number of days to complete, during which they structure their own time in completing the notes and guided exercises. These notes break down concepts into digestible chunks, are written in a very informal manner, and after giving students information about a given topic, direct students to complete a guided exercise in a textbook. The guided exercises provide students with a step-by-step process intended to help students internalize and practice the topic covered in the electronic notes. Generally, I will allow the class one day with the notes, and then the following lesson will be a more traditional teacher-led lesson, approximately 30 minutes in length, where I will highlight the major key concepts in the chapter by creating a program or two with the students. Other times, in lieu of a guided textbook exercise, the notes may direct students to watch a tutorial on YouTube or some other web page, and to follow the directions in the online tutorial so they may practice the topic they just read about. In this manner, my classroom is largely student-centered about 85-90% of the time, using my structured notes, while I circulate address individual student questions. The other 10-15% of the time is a more familiar and traditional teacher-led routine. Once students are comfortable with this new level of “teacher-supported freedom”, things move along very smoothly, and students gain confidence in their abilities to help one another, and in their ability to learn “on their own”.

In this manner, an entire topic in a traditional textbook can be covered. I have also chosen or created a list of unguided exercises that students are to complete at the end of each note package. After a few chapters of notes and exercises are complete, we stop doing notes for a while, and spend 5-10 classes completing a project (Appendix E). The project is designed to bring together all the topics students have learned, usually to create some sort of game. Figure A below will help you visualize the structure of the course.

Activities that Support Learning.

There are a variety of activities that are completed that help students think about their learning process.

Reflection Paper, Start & End of Course. The beginning of the course utilizes a reflection paper (Appendix F)., which students use to plan goals, let me know their background information, and why they took the course. They will receive this paper back at the end of the semester and reflect on what they have accomplished, learned, and whether or not they wish to continue coding in the future and the reasons why or why not.

Pair Coding. Approximately two to four times a month, I will hand out instructions to complete some sort of specific program to solve a given problem or situation. Students are paired up (best pairings are similar skill-level students together, or also high competence with lower competence works well) to complete the assignment utilizing the same computer. One student controls the keyboard and types the code, the other student is supposed to constantly review the code and catch errors as they arrive. The students are supposed to switch roles at least one time. In the coding industry, this is known as *pairing* or *pair programming*. Several studies have been completed in regards to its effectiveness in both industry settings and learning settings. If done effectively, this sort of coding practice can help students with their confidence in coding, and can help students catch their own coding errors as they code on their own (Williams, Kessler, Cunningham, & Jeffries, 2000; Williams & Upchurch, 2001).

Teacher-Student Interaction & Mini-Lessons. By no means are you able to put up your feet and relax due to the nature of the course. To the contrary, many students will have questions, and may require that you explain concepts from the notes to them. This takes two forms: teacher helping directly at the computer, or pulling small groups of students aside for short mini-lessons on the whiteboard or projector. In this manner, students get the help they need, when they need it, in the context of the problem that requires the information. This differs from traditional direct-instruction, which is essentially a large information dump each time, usually out of context, and too quick or too slow for most learners. You can expect to teach several mini lessons each week.

Student-Student Interaction. This takes some time to foster, but if you repeat every day to students that they should be "good neighbours" and nice coders, eventually you will find that students will become comfortable asking their neighbours for assistance. Watch out for students who are just copying code. You can and should address this in your class or lectures both by modelling the behaviour and with some videos to spur some dialogue about expectations (Appendix G). Make sure that you are mindful of your own teacher-student interactions, these will help model for students how to properly and effectively give assistance. In time, the payoff is huge, as your time will free up to help other students, and students will have access to a variety of resources when they are stuck in their code.

Assessment of Projects and In-Class Work.

In a standard unit, students are instructed to hand in a particular piece of code on the agreed upon last day to work on the unit (assessment of such can be found in Appendix H).. As time moves on, students do tend to pick up different aspects of the course material. It is both a strength and drawback to teaching like this. To ensure some regularity, at the end of every unit of notes, students will study a practice theory test, in which I have purposefully selected all the items I wished them to learn during the unit, in order to help address any gaps in their learning up to this point. We agree upon a "theory day" as a class, and the students write an online theory assessment for that unit, which usually also includes them writing a small program that demonstrates their abilities.

Assessment of Projects.

There are a few major projects, generally games that we complete throughout the course. I encourage students to add their own unique game elements to a project, and account for this in the regular rubric I use (Appendix E). Basically a student can achieve an 80% for completing the project to my specifications, and the other 20% allows them to show me their creativity, and flex their independent student muscles by adding new game elements. This is part of my scaffolding strategy, to get them used to the idea that in real life, or at least in the coding industry, you decide how you will complete a project, and after some basic guidelines such as what the project should do or act, it is up to you to decide *how* it will execute, and what other features you wish to include in order to make a stellar product that people will want to use.

Dealing with Student Diversity on Projects.

Some students will be ready to by the first project to propose their own project idea; otherwise they do a pre-canned project that I have created. I have a template that students use to help them plan (Appendix A), and for me as their teacher to ensure their goals can be attained in our timelines, and to also prepare me for the sort of support I may have to offer the student in achieving their goals. The planning document includes sections for conditions for success, and conditions in which the student may have to re-evaluate whether or not they can achieve their goal. Feedback from previous students indicates they appreciate this opportunity to plan in such a manner. It allows them to introduce some structure in their endeavour. In the grade 12 course, this template becomes the norm for how students plan their projects, as they completely choose their own.

Assessing these projects is something done on a case by case basis. I generally start with my general rubric, and using the planning document the student made, I discuss with the student what elements of their project are important enough to include in the rubric. There are times where the general rubric can be used for these unique projects without much modification, but many times items must be added and removed. The added items tend to be outcome based; items such as the program works in the intended manner, is free of bugs or glitches, are generally included in the rubric. It is decided ahead of time how many points out of 100 each item should be worth—this also get the student thinking about priorities, and what is truly central to the success of their program.

Structure.

The first 5-8 classes are more heavily weighted in direct instruction methods. Generally, the first 2 or 3 classes are 100% direct instruction, the next 2 or 3 classes are 25% direct instruction, and 75% guided learning This is required as many students have no prior knowledge or experiences in coding to draw upon. The teacher should use this time to explain the goal of the course and that students will learn how to become self-directed learners. These expectations are made clear.

Most of your teaching time is spent travelling around the classroom and answering individual questions from students. This allows you to address any students who are following falling too far behind, and to get a sense of what concepts the class is struggling with. If I notice two or three different students asking the same conceptual or procedural questions, I will do a quick mini lesson for students who are interested, or with the entire class.

The Grade 11 and Grade 12 Courses

Appendix C and D give you an idea of the structure of the grade 11 and grade 12 courses. In Grade 11, we spend about half the class time learning how to use Java using much the same methods as outline for the Grade 10 course. The major difference here is that during the second half of the course, we spend all our time completing projects chosen by the students, who will use Appendix A to help themselves plan. For students who still are a bit reluctant to plan, or just want some ideas from me, I have a variety of projects that I can give them (Appendix I includes a few samples). Or I will sit down with the student, and using their beginning of course reflection (Appendix F), I may help them identify projects that align with their goals.

The Grade 12 course is entirely project driven, with all projects being chosen by the students. There is one project that I require my students to complete, which revolves around teaching others how to code. Before I get to that, I will briefly outline why I think teaching is so important in this course. If one can teach something, it is usually a pre-requisite that they understand the material taught on a very high level. All the planning and creation that is involved in creating a lesson, and then teaching, exactly mirrors some of the major skills that I think are important for my students both as programmers and important life long skills. Thus, this teaching initiative serves my course purposes well, and works to spread awareness about computer science in elementary level schools. This initiative involves having the Grade 12 students teach themselves a programming language, Scratch, which was designed by the MIT Media Lab, to help people with no coding experience learn how to code, particularly children. It uses an easy drop and go block based language, which is very visual, and allows the user to relatively quickly create games that work using sprites.

I arrange for my students to teach between five and seven lessons in a few grade 7/8 classes in our school division. I do this by reaching out to nearby teachers in elementary schools, and asking them if they would like to have some free computer science lessons for their class. I have found the response to be very high. I coach my students in creating lessons, and give them time to practice with my feedback. They design their own lessons, and then I accompany them on the first trip out to the grade 7/8 class, and afterwards they are supervised by the grade 7/8 classroom teacher and go by themselves. These lessons are spread out every seven to ten days, and we typically do two schools in a semester. My Grade 12's have so far identified this as a very valuable experience for them, as it deepens their understanding of basic programming concepts when they explain it to others in such a formal setting as a classroom lesson. It also builds their confidence in their abilities to think quickly on the spot and answer questions.

Otherwise, my Grade 12 course is driven by student created projects. See Appendix I for a list and brief description of some of the projects my grade 11 and 12 students have completed for an idea of the diversity that students can dream up.

The challenges of the student-centered model

Boy, this is going to be a long section. I never once promised this method is a silver bullet that would make everything easier—to the contrary, I will state again that while the benefits are great, the road there is perilous.

According to Lazonder (2004), the fatal flaw in true open-ended inquiry-based learning—the ultimate goal where I aim for students to arrive at by the time they begin the grade 12 course—is that students will fail to infer trends about their learning from the "data" of how they are doing, they will have trouble systematically attending to the gaps in their learning, and recognizing when they will need to attend to anomalous performances in their learning. However, studies by

Mulder, Lazonder, and De Jong (2010) show that inquiry-based tasks are possible if students are adequately scaffolded throughout the entire project. There is much debate into what this support must look like, when and where it must be implemented, and in what forms it needs to be implemented (Lazonder, 2014, p. 454). For example, some researchers insist that students should have all the required skills and knowledge prior to the inquiry project (Kirschner, Sweller, & Clark, 2006, as cited in Lazonder, 2004, p. 454) and others believe a "just in time" approach is suitable (Hmelo-Silver, Golan Dunca, & Chinn, 2007, as cited in Lazonder, p. 454).

Thus, it is key to scaffold students all throughout the process. By no means is this an easy task—in fact, I regularly find this class much more work to manage and run than say my more direct instruction style in a Grade 12 Pre-Calculus class—however the benefits are huge.
So the main challenge to a student-centered classroom that makes use of inquiry lies in reorienting both the teacher and the students to their new roles. The teacher must learn how to let go of classroom control, while at the same time ensuring that the classroom remains a controlled environment where learning can occur (read: scaffolding!). At the same time, the teacher must create notes that are entirely student friendly, so students can successfully learn on their own. Most textbooks are not written in such a manner, so it is up to the teacher to do this. I have found that a student-centered model of learning is more front end heavy: it takes time to compile and create the required resources, but afterwards, the teaching is easier in this respect, since you are not required to keep a captivated audience for 65 minutes every class. Also, since you will be interacting with individual students on a much more frequent and longer basis that in a traditional teacher-led classroom, this is another challenge, as you will generally be answering for the entire class period.

I found that the general advice that the teacher does not need to be a master of the material students are learning is not necessarily true. In order to give effective formative feedback for example, teachers must become at least somewhat familiar with concepts students are attempting to learn. Ruiz-Primo (2011) found that effective formative feedback can only come from teachers' who "possess both sufficient content and pedagogical knowledge in order to interpret and spontaneously act upon students' contributions" (p. 18). To this end, while I believe more than ever that a teacher's role in the classroom is now involving a bit more on guiding the social process of learning (Muller, 2014), teachers must still have strong content knowledge and teaching knowledge in order to be effective.

In an environment where students propose their own projects, teachers can easily be overwhelmed by the variety of topics that students will require resources for. This can be overcome, but teachers must be willing to learn new things, often at the same time as the students are learning them. In time, as your knowledge base expands, it becomes easier and easier. You may also find that within your class certain students will begin to become quite proficient in helping others and will expand their knowledge base as well, allowing other students to tap into their peers’ own unique knowledge and contribute in that manner to the classroom environment.

The students at first struggle with the concept that they are now responsible for their own learning. In Grade 10, I find that the average student takes about 5-15 classes to fully grasp the structure of the course, and to become independent enough to comfortably move through the resources I have prepared for them to learn. Students also struggle at first with what to do when they are stuck and cannot access their teacher immediately. Routines and techniques must be both modelled and taught to students so they clearly understand how they should proceed in this situation. Some students are also not used to holding themselves accountable for their own learning, and struggle with this initially. Other students have not yet mastered how to read about a concept in text form, or watch a video about it, and then transfer it to a guided learning lesson afterwards. Skills that must be taught are how to electronically search their notes when they forget concepts, and when to recognize that they need help from a teacher or another student. (see Table 2 below for a summary of this section).

Table 2. Summarizing some the challenges for this model for the teacher, student, and also assessment challenges.

Teacher Challenges	-Teaching the whole class at once to more individual and small group lessons as needed. -Recreating notes and other learning resources so they are more student friendly in an environment where teacher guidance is on demand versus all the time. -Adjusting classroom management to support individual and small group student learning. -Developing a common core of knowledge that all students must learn in order to become successful in the course; or the common knowledge that a professional in that field should acquire before specialization. -Becoming comfortable with the idea that many students will come out of the course with a different knowledge base, depending on what projects they chose to pursue. -Answering questions for extended periods of time can become exhausting. Finding a balance between when direct instruction is more effective, and when small group or individual instruction is beneficial is daunting at first. -Becoming a master of a variety of different topics you may not be familiar with, and having these topics potentially change year to year.

Teacher Challenges	-Teaching the whole class at once to more individual and small group lessons as needed. -Recreating notes and other learning resources so they are more student friendly in an environment where teacher guidance is on demand versus all the time. -Adjusting classroom management to support individual and small group student learning. -Developing a common core of knowledge that all students must learn in order to become successful in the course; or the common knowledge that a professional in that field should acquire before specialization. -Becoming comfortable with the idea that many students will come out of the course with a different knowledge base, depending on what projects they chose to pursue. -Answering questions for extended periods of time can become exhausting. Finding a balance between when direct instruction is more effective, and when small group or individual instruction is beneficial is daunting at first. -Becoming a master of a variety of different topics you may not be familiar with, and having these topics potentially change year to year.

Student Challenges	-Navigating a new course structure with guided notes, some direct teacher instruction, and micro-teaching can be difficult. -Becoming more responsible for their own learning can be a challenge at first. -Learning to recognize when teacher or peer guidance is required, and how to manage time while guidance is on its way. -Learning how to rely more on their own peers, and becoming a source of knowledge for their own peers, can be daunting at first. -Becoming more reflective in understanding if they have understood a concept without immediate, direct teacher input. -Learning how to effectively interact with other peers to receive help. -Learning how to effectively help their peers when asked a question. -Time management.
Assessment Challenges	-Creating fair rubric and assessments for student-proposed projects in grade 10-12. -Having students identify what aspects of their program are important enough to be included in the assessment can be a slow process, especially when students may not be aware of what these elements may be until they have gotten a good start on the project. -Providing formative feedback for each student during class time is crucial—and thus time consuming given how many different student-proposed projects can be going on. -Assessing a variety of student products that are all different.

Perceived Benefits

Herrington, Reeves, and Oliver (2014) speak to authentic learning environments and how "meaningful learning will only take place if it is embedded in the social and physical context within which it will be used" (p. 402). This was one of my guiding principles in developing the courses in the way I have. In my experience, it would seem that disengaged learners often do not learn anything very deeply or meaningfully. This idea is echoed in a review of research by Ruiz-Primo (2011), who states that "disengagement has a strong negative impact on achievement" (p. 22). Luckily, with the structure of this course, I have rarely observed disengaged learners. When you give students charge of their own learning, and provide them with enough scaffolding that they feel supported, disengagement will not be a problem. This can be the way you take stock of how your course is doing--are students engaged in their work? Are they choosing unique tasks for themselves to complete? You will find that once students are comfortable enough to propose their own projects and complete their own inquiry into the new things they need to learn to accomplish their goals—you will have a room full of engaged learners.

In their year-end reflections to me, both the grade 11 and 12 students note that their most valuable experiences in the course were teaching their peers, and being able to interact with one another, and me their teacher, and having their notes and resources available to them electronically. In the case of the grade 12 class, who go out and teach elementary school children the coding language Scratch, these students note that this experience helped them the most in actually understanding the subject material in a real way (note: this project is another of my scaffolded "mandatory" projects that sort of have an inquiry flavour to them, as the students must teach themselves the language and create their own lessons for the elementary aged children).

I can also point to the case of my grade 11 class last year. After we had completed my notes and projects for Java, as a class, we decided that we wanted to learn Visual C#, another programming language. I facilitated this by choosing the topics we'd learn, and then allowed groups of one or two to choose a topic. Each group/individual student created electronic notes for their topic, exercises to practice, and a teaching presentation on their topic. Each group taught the class, administered the exercises, graded an exercise, and answered all the questions pertaining to their topic. This entire process took us over 4 weeks. I had another student who created a "game theory unit", and created some really excellent notes and tutorials in how to make games where objects moved (such as space invaders), how to create your own animated characters fairly easily with sprites. I allowed him to use another student who was interested as a "test subject" for his notes--she would try to follow them and give him suggestions as to whether or not they were clear enough. I plan to make these notes part of my regular teaching regime in the grade 11 course (or, of course, since I try not to prescribe more than 50% of the material in grade 11--it may be an optional unit for those who can't decide what to do).

It is no surprise to me then that students report to me a great increase in their confidence in their abilities to learn on their own, to plan projects, and to code. This may also enjoy a touch of

personalized learning—while students do not choose everything they get to learn until they arrive to the grade 12 course, instead, I scaffold them starting in the grade 10 course to eventual full personalization and student-led (mostly—I only help them in the planning, and ensuring they have their resources) inquiry. If I have done my task correctly, research has found that students show a richer conceptual understanding of topics learned using inquiry compared to passive or receptive strategies (Minner, Levy, & Century, 2010, as cited in Lazonder, 2004, p. 454).
It is interesting to note that not all students appreciate or even enjoy choosing what they learn in grade 12—some students (a minority of them at this point) request that I choose what they will learn. Either way: the student has "chosen" what they wish, whether they choose it themselves or prefer for the teacher to choose for them.

Conclusion

And there it is. Perhaps I can leave you with one comment from a teacher peer of mine, who preps nearby where I teach my grade 10 course. She said to me "hey, do you ever actually teach anything in this class?" Of course, this comment was actually well-meaning, as she also noted that my students appear to also be working and accomplishing "something computery" (in her words). It wasn't really until this moment that I had realized what I have done. A classroom where I spend most of my time interacting directly with students one-on-one, or in small groups, was a classroom where apparently the most work was getting accomplished, with students engaged in their learning because they feel they have a measure of control over how they learn it, why they learn, and when they learn it. I still manage to do some direct instruction about 15% of the time. And yet, my colleague noted that she rarely sees me directly teaching the entire class at once. This leads me to my concluding point.

Most technologies available to teachers at present make tasks faster, but also have the potential to make learning tasks more social, and can connect us to the world outside the walls of the classroom. I think we would do well as teaching professionals with students' best interest at heart to embrace this fact. As Seymour Papert (2003) might have said, this fact is truly fundamental to the topic of education today. My grade 12 students all learn different things by the nature of the course where they choose their own path… while this is scary to some, (certainly I was nervous of this idea when I first started), Papert challenged the notion that "everyone should have the same knowledge set and acquires it in the about the same order" (p. 11). This idea seems well suited to this particular subject area of computer science.

Now more than ever, I believe we are at a cross-roads with what exactly teacher's roles will be in the traditional classroom. I can think of no better concluding thoughts than Muller's (2014) comment that in order to really spark learning in the classroom, we must embrace "a social environment, with other learners, with a caring teacher" (location 6:57).

References

Edwards, C. (2013). Self-paced mathematical instruction. *Mathematics Teaching in the Middle School, 19*(4), 230-236.

Herrington, J., Reeves, T., & Oliver, R. (2014). Authentic Learning Environments. In Spector, J., Merrill, D., Elen, J., & Bishop, M. (Eds.), *Handbook of Research for Educational Communications and Technology* (4th ed., pp. 401-412). New York, NY: Springer.

Lazonder, A. W. (2014). Inquiry Learning. In Spector, J. M. Merrill, M. D., Elen, J., & Bishop, M. J. (2014). *Handbook of Research on Educational Communications and Technology* (4th ed.). New York: Springer.

Mulder, Y., Lazonder, A., & De Jong, T. (2010). Finding out how they find it out: An empirical analysis of inquiry learners' need for support. *International Journal of Science Education, 32,* 2033-2053.

Muller, D. (2014). This Will Revolutionize Education: August 22, 2017 [Video file]. Retrieved from https://www.youtube.com/watch?v=GEmuEWjHr5c

Papert, S. (2003). What are you talking about? A Dialogue with Seymour Papert. *Convergence, 9*(10)*,* 10-12. doi: 10.1177/135485650300900202

Ruiz-Primo, M. (2011). Informal formative assessment: the role of instructional dialogues in assessing students learning. *Studies in Educational Evaluation, 37*(1), 15-24. doi:10.1016/j.stueduc.2011.04.003

Williams, L., & Upchurch, R. (2001). "In support of student pair-programming". *ACM SIGCSE Bulletin*. **33** (1): 327-331. doi:10.1145/366413.364614

Williams, L., Kessler, R., Cunningham, W., & Jeffries, R. (2000). "Strengthening the case for pair programming". *IEEE Software*. **17** (4): 19–25. doi:10.1109/52.854064.

Zion, M., & Mendelovici, R. (2012). Moving from structured to open inquiry: Challenges and limits. *Science Education International, 23*(4), 383-399.

Appendix A

Project Proposal And Planning Document

This is how I help guide my students when they decide they are ready to tackle a project on their own. I will scaffold the class by directly going over with them how this document works, and often we will do one as practice. This is a back and forth process—expect your students to submit this to you, and expect to submit it back to them with comments and changes they need to address before you look at it again. The major stumbling block is students need some help in developing their ideas, and communicating them effectively on paper. You must help them see the value in this process—so make it valuable to them!

New Project Description	
What sorts of new things will I have to learn *(list each topic or item)*	
How will I learn these things? *(what sort of practice do you need to complete to learn the items above? Do you just need to read about them, or create some simple programs, or can you apply it directly to what you already know?)*	
What resources do I plan to use in learning these new things	
How long will each item take me to learn or complete	
What will I do if I get stuck on something? *(Or*	

think: what are my expected barriers or difficulties and how will I address them?)	
Conditions for Success *(how will I know if I'm successful in achieving my goals? In other words: when do I know "I've successfully implemented this idea"?)*	
Conditions under which I need to re-evaluate or quit what I'm doing *(How will I know when I need to re-evaluate my goals, or get rid of a goal? … Remember, this is done to help you manage your time effectively… good CEOs and leaders know the conditions under which they need to "pull out" of an idea or re-evaluate goals …)*	

Appendix B

Grade 10 Computer Science Course Outline

Teacher: Mr. M. Di Muro

Please feel free to contact me if you have course-related questions at dimuro.matteo@bsd.ca

Course Description/Objectives

This course will introduce the student to the concept of computer programming using Microsoft Visual Basic. These topics will be covered at an introductory level only and will provide valuable problem-solving skills. Computer Science is generally a basic requirement for many college and university level programs/degrees.

One major theme of this course will be learning how to "help yourself". Many jobs require that you continue to learn, often with very little support from an "instructor". You will also often be required to work well with others, and project plan on your own. To help you get there, I have set up this course in a way that you can work at your own pace, but within a given time frame I have created for each unit. This means that you need not cover a certain topic in one class, but you must cover it within the time frame given. This gives you time to get comfortable on your own, within reason. Communicate with me if you are having difficulties!

Students are encouraged to work ahead as well. If you feel comfortable, and you are ahead, it is possible to create your own project ideas, and I can help you with that (this is more aligned with what we do in the grade 11 course).

We will examine how to use resources online to help you learn how to code, how to evaluate a good source from a "bad" source, and how you organize yourself in a way that should ensure your success in this class. I highly encourage students to work together in class, we are a community of learners who are all trying to learn the same topic—be nice, and help your peers when you are able to! Not only will this reinforce concepts for you, but it will give you a great feeling of accomplishment!

We will begin to develop your skills in planning, creating, and implementing your own ideas within the projects we will do together as part of the course.

Topics of Study:

Unit	Outcomes
Chapter 2: Introduction to Visual Basic (~8 classes)	• Become familiar with VB IDE (Integrated Development Environment) • Understand the need of and become proficient in commenting code • Manipulate property values and menus of forms, command buttons, labels and images
Chapter 3: Variables and	• Differentiate between different data types (strings, integers, double, constants, currency) and understand when to use each, declare

Constants (~10 classes)	variables • Understand syntax errors and run-time errors • Utilize "pseudo-code" to understand and solve a problem and use good programming style guidelines
Chapter 4: Controlling Program Flow with Decision Structures (~15 classes)	• Use: If statements, If…Then statements, nested If..Then…Else statements, If…Then…ElseIf statements, and use Select…Case Is statements • Write Boolean expressions and understand static variables • Understand logic errors and design problem-solving strategies • Use message boxes, counters, and check box objects
Chapter 5: Controlling Program Flow with Loops (~11 classes)	• Use Do…Loops and Infinite Loops, explain how infinite loops can occur • Use input boxes in applications, utilize String Concatenation • Use String class and its methods (LetterCount, StringTest, FindString, Compare)
Chapter 7: Using Files (~5 classes)	• Use files for data input • Create, copy, and delete files at run time • Understand file streams and use a stream reader to read file contents • Use a text box to display file contents.
Chapter 9: Color, Sound, and Graphics (~10 classes)	• Apply color to an interface and include a Color dialog box in an application • Add sounds and images to an application and distinguish between image and sound file formats • Use the Graphics class and its methods • Create event procedures that respond to mouse events and use a Timer control to create animation
Putting it all Together (~8 classes)	• Compile everything you know into a grand masterpiece of programming genius and chose a project from various pre-approved project ideas, and add your own personal flare if you wish. • Prepare your brain for the biggest written exam ever ☺

(please note that order of units completed will vary and/or be combined). All units will be evaluated with a variety of assignments, checkpoints, case studies, tests and/or projects.

Course Evaluation Structure

Several types of evaluation will be used in the course. This will allow students to display their level of learning in a variety of manners and also expand their skill levels in different presentation methods.

- Tests 25%
- Summative Assignments/Projects 40%
- Check Points/Reflections 15%
- Final Assessment 20% (10% written test, 10% final project)

You will be provided with regular updates of your cumulative mark. If, at any time, you have questions about your mark, please see me immediately. Many parents are registered on **Vincent Massey's Home Logic** program, which allows parents and students access to their updated marks from their home computer. If you are interested in becoming a Home Logic user you can contact the main office at 729-3170 or email Mrs. Val Smith at smith.val@brandonsd.mb.ca

And now, for the "mandatory" part of every course outline as set out by BSD policy:

Absent and Test Procedures

It is the student's responsibility to first get missed notes/materials from a classmate or my website and then come and discuss the notes and assignment with me.

All missed tests will be written on the first day back, unless arranged with the teacher. There are **no test re-writes**. If I know ahead of time, I will do my best to accommodate the student with prepared notes and assignments ahead of time. *Please refer to the Student Responsibility Guidelines for Assessment and Evaluation regarding absent and late assignment/test policy*

Expectations/Rules for classroom:

1) **Respect** is the main rule in my classroom. I strive to create an environment of mutual respect in which learning can take place. This means students are expected to respect themselves, their classmates, and teachers; both in the classroom and outside the classroom.
2) You may use your cell phone as long as it is appropriate, and you are not distracting yourself or others!
3) We work in a computer lab! So please do not bring food or drinks, other than water to the computer area. Our room has a small area away from the computer that is designated for use in cases where you must eat a snack… please ensure you clean up after yourself and WASH YOUR HANDS before returning to the computer area!

Appendix C

Grade 11 Computer Science Course Outline

Teacher: Mr. M. Di Muro

Please feel free to contact me if you have course-related questions at dimuro.matteo@bsd.ca

Course Description & Objectives

This course is meant to continue and build upon what a student learns in COMPSCI 20S. The main objective of this course is twofold: (1) to expose students to Object-oriented programming languages, namely Java. (2) to have students take initiative in planning, creating, and evaluating their own projects.

One major theme of this course will be learning how to "help yourself". Many jobs require that you continue to learn, often with very little support from an "instructor". You will also often be required to work well with others, and project plan on your own. To help you get there, I have set up this course in a way that you can work at your own pace. I will give you suggestions as to where you should be in the course material, but it is up to you. Since you already know a computer language (Visual Basic), you will find that learning another language such as Java will not take you nearly as long as it took you the first time. The units of study below should take you approximately half the course to complete, leaving the second half of the course open to you completing projects I have laid out for you, or projects that you develop on your own.

We will look at how to use resources online to help you learn how to code, how to evaluate a good source from a "bad" source, and how you organize yourself in a way that should ensure your success in this class. We will develop your skills in planning, creating, and executing your own projects that align with your own interests and goals as an individual. Assessment of these projects will be discussed between you and I, as you highlight what aspects of the project are essential and therefore should be graded.

Java is a widely-used computer language and is often offered as a first year course in many universities across Canada.

It is recommended that students have a mark of at least 70% in COMPSCI 20S and a strong background in Mathematics (Pre-Calc or Applied) as mathematical concepts will be regularly utilized within this course.

Topics of Study:

PART I: Half the course time will be spent completing:

Unit	Outcomes
Chapter 3: Introducing	• Introduce object-oriented concepts (p 35) • Objects, classes, and packages (p 35-36)

Java	• Creating and executing a Java application (p 37-38) • Application output (System.out.print/println) (p 39-40) • Code conventions (p 41) • Algorithm design (p 42)
Chapter 4: Variables & Constants	• Primitive types vs. objects (p 51-52) • Variable declarations (p 52) • Classes (abstract data type) (p 54) • Simple data types (int, boolean, double) (p 53) • Java packages (p 54) • Obtain a value from the user (p 55) • Numeric expressions (p 57) • Type-casting (p 58) • Constant declarations (p 61) • Syntax, run-time, and logic errors (exceptions) (p 62) • Read and understand a problem description, purpose, and goals (p 64)
Chapter 5: Conditional Control Structures	• Conditional control structures (if, switch) (p 77) • Generating random numbers (Math.random()) (p 81) • Logical operators (p 83) • The Math class (Math.abs(), Math.pow(), Math.sqrt()) (p 85)
Chapter 6: Loop Structures & Strings	• Iteration (while, for) (p 101) • Counters and accumulators (p 103) • Debugging techniques including using a debugger, adding extra output statements, and hand-tracing code (p 106) • The String class (p 108)
Chapter 7 Methods:	• Methods (p 127) • Top-down development (p 127) • Procedural abstraction (p 127) • Method declarations (p 129) • Parameter declarations (p 130) • Pre- and post-conditions (p 134) • Identify boundary cases and generate appropriate test data (p 137)
Chapter 8: Classes & Object-Oriented Development	• Data abstraction and encapsulation (p 149) • Design and implement a class (p 150) • Functional decomposition (p 150) • Encapsulation and information hiding (p 152) • The Object class (p 155) • Inheritance (p 156) • The has-a class relationship (p 157) • Object-oriented development (p 161) • Extending a class (p 181) • Interfaces (p 193)
Chapter 10: Arrays	• One-dimensional arrays (p 207) • Traversing an array (p 208, 221)

	• Array element insertions and deletions (p 208) • Arrays with meaningful indexes (p 210) • Sequential search (p 214) • Two-dimensional arrays (p 215) • The ArrayList class (p 220) • Wrapper classes (p 221) • Autoboxing and auto-unboxing (p 223)

Part II: Projects, Projects, Projects (Student Choice or Teacher Provided)

As described in the course description, you will spend the last half of the course either completing projects set out by me, or, preferably, you will plan and create your own projects with my assistance.

<u>Course Evaluation Structure</u>

Some topics have a final theory test and case program you must complete. You will also complete a final theory test for Java when you are complete the first half of the course. This is to ensure some regularity in what you have learned about Java.

Assessment of student created projects are on a case by case basis, which will be created by your teacher and you working together. With the teacher's help, you will identify key elements of your project that are important and thus should be included in the assessment of the completed project.

- Tests 20%
- Final Theory Test in Java 10%
- Planning Documents 10%
- Summative Assignments/Projects 30%
- Check Points/Reflections 10%
- Final Project 20%

Appendix D

Grade 12 Computer Science Course Outline

Teacher: **Mr. M. Di Muro**

Please feel free to contact me if you have course-related questions at dimuro.matteo@bsd.ca

Course Description/Objectives

This course is meant to continue and build upon what a student learns in COMPSCI 30S. The main objective of this course is help students do several things:

1) Learn how to identify their own goals, plan to execute them, plan on how they will know if they have achieved these goals, and then follow through.
2) Choose a computer language they wish to learn, and learn it.
3) Create their own projects, either with their peers or individually, that have a real life application or value. This may involve coordinating with other classes, other teachers, outside experts, and writing letters asking for permission to do these things to the school principal.
4) Teach others the value of computer science by taking initiative and teaching the Scratch programming language to local area grade 7/8 students in the BSD.

One major theme of this course will be learning how to "help yourself". Many jobs require that you continue to learn, often with very little support from an "instructor". You will also be required to work with other people regularly in this course—a vital skill in the future, as few large projects are ever solitary endeavors. I can and will give you suggestions as to where you should be heading, whenever I feel this information may be beneficial to you. But largely, within the structure of the course I have built, many choices are left up to you as long as you are reporting them to me and we engage in useful conversations so we all share a common understanding.

We will also continue to examine how to use resources online to help you learn how to code, how to evaluate a good source from a "bad" source, and how you organize yourself in a way that should ensure your success in this class from a project planning and execution perspective.

It is recommended that students have a mark of at least 80% in COMPSCI 30S in order to feel comfortable in this course.

Topics:

Remember, for the most part, I will be guiding you in planning and executing your own projects. However, there are common elements to the course that we will complete together, they are:

- Start of course survey & reflection and goal planning
- Teaching computer science: Spreading awareness with Scratch *(this is an on-going process throughout the course, as we will be teaching in local area schools)*

- Overview of Planning & Goal Setting Documents
- Project Creation: *how to choose, identifying resources, goals, roadblocks, conditions for success, how you will know if you accomplished your goal (hence: assessment), when to modify/leave the goal*
- Choose a language—Learn it
- Create & execute your projects
- Reflections on your projects

Course Evaluation Structure

Several types of evaluation will be used in the course. This will allow students to display their level of learning in a variety of manners and also expand their skill levels in different presentation methods.

- Summative Assignments/Projects 60%
- Check Points/Reflections 10%
- Planning Documents 15%
- Teaching Initiative 15%

Appendix E

Marking Guide for First Major Project—Grade 10 Computer Science

Students create a "pick your own adventure" type game, where the user navigates the student created story by using a variety of buttons on the form, with each new part of the story represented by a new "screen" on the form. Students first start the project by creating a storyboard, which I check before they begin their coding. Students are also required to add a new unique element to the game—this is reflected in the rubric. This causes them to teach themselves something new by using resources in our textbook or on the web, or their own peers.

Category	5	4	3	2	1	0
Story & Story Length	-Student has programmed at least 5 forks, and perhaps has added some sort of other story device to increase length of game. -The story makes sense, spelling and grammar are perfect, and it is evident that thought went into the story planning. Game flow is flawless.	-Four forks are present in the story -There may be a 1 or 2 spelling and/or grammar issues, perhaps a small inconsistency in the story or flow of the game.	-The minimum three forks have been programmed successfully. -3 spelling/grammar errors, or a story that is somewhat convoluted or doesn't make sense in some spots.			-There are 2 or less forks, and/or forks are not working properly in game. -6 or more spelling/grammar errors, and/or story does not make sense or is incomplete.
STORY BOARD	Storyboard is complete, neat, organised, and clearly shows the entire story in a mapped out process which would have aided you in creating your game.		Storyboard is complete; however, it does not clearly map out the story and the options the player can select.			Storyboard is missing, or is incomplete.
Coding Style & Comments	-comments are present above each click event with brief description of what event executes -Coding style is perfect. Initiative variable names & object names -Code is well formatted and clean	One of the following: -Comments are present but may not always be clear -Coding style is overall very good, but 1 or 2 variable names or object names are unclear -Code may have 1 or 2 spots where not formatted well	Two of the following: -Comments are present but may not always be clear -Coding style is overall very good, but 1 or 2 variable names or object names are unclear -Code may have 1 or 2 spots where not formatted well			-Comments are not present and/or not clear at all -Coding style: variables/object names are not initiative -Code formatting is lacking completely
Program Execution	-The game works without any bugs or errors.	-One small bug or error in program	-One bug or error that may effect game	-Three bugs or errors that may effect	-Four bugs or errors that may effect	-Major bugs or errors in code that make game

		execution	playability, however game can still be completed	game playability, however game can still be completed	game playability, however game can still be completed	unplayable
Effort/ Make it Yours/	-Project is very polished, (think: would I buy this?) -It is apparent that the project has been tweaked to look and work differently from others -the student shows a level of pride in their work by adding new game elements that were not required in the directions. -Student added extra features to the game that match the game's theme or add to the play value of the game, or learnt how to use code not covered previously in class.	-Project is very polished, (think: would I buy this?) -Two or more bullets from the level 5 column w successfully completed	-Project is very polished, (think: would I buy this?) -One bullet from the level 5 column completed.	-Project may not be as polished and professional as it could have been -One bullet from the level 5 column		There is no evidence that anything beyond the minimum requirements has been attempted.

Total: 25. Teacher Comments (The Bad, the Ugly, the Good! Time for BUG!):

Created by Matteo Di Muro

Feel free to use/modify as long as you credit me!

dimuro.matteo001@gmail.com

Appendix F

Start and End of Course Reflection Papers

GRADE 10/11/12 START OF COURSE RELFECTION PAPER

Research shows that one key difference between super successful people and non-super successful people are that successful people WRITE DOWN and SET goals that they wish to achieve. They also write the conditions under which they would retire that goal or look at modifying it. We will work more on this as we progress. In the meantime, I'd like you to identify your current knowledge base, and begin thinking about goals. You will be able to change your goals later, so please write some down now, and realize you are not necessarily stuck with these goals as we progress through the course, we will do this more formally in a few weeks:

1) What is my current knowledge base? Include in computer science but also include other topics you are knowledge in.

2) You are taking a course in computer science that has a more "open-ended" structure than most courses you may be familiar with. I'd like you to think about setting some goals in this course. What sorts of programs do you want to make? Do you want to learn a different language? Sky is the limit.

3) What sort of support do you think you will need from me as your teacher? I don't know everything, but I CAN get you resources so you can find what you need.

4) By what metrics will you evaluate that you achieved your goals? For example, if my goal is "to learn how to code using Java", how will I know when I have achieved this? I might write: "I will learn how to make a program that does ______ in Java, and when I have I will have achieved my goal". Or I may list certain topics I wish to become proficient in (I will write 10 programs that make use of arrays in Java may be a viable goal).

GRADE 10/11/12 END OF COURSE RELFECTION PAPER

Cognitive studies have shown that besides content knowledge, reflection (thinking about your thinking, and evaluating experiences after they have occurred) is one of the keys that separates a true master and an "amateur" in almost any field or profession. It also happens to be a great way to learn. Therefore I would like you to write me a reflection paper about your experiences in this course.

You will compose a two-page (double spaced, size 13 font… you may write more than 2 pages if you wish) reflection on…

- □ How you spent the course (what projects did you work on? Where there projects that you attempted that were not successful? Often failure is a better teacher anyways… what did you learn during these failed projects?)
- □ What you have learned (whether it be a programming language, or specific skills)
- □ Skills you were able to develop or expanded on in this course. Make sure to talk about your experience teaching Scratch to grade school children and adults
- □ What sorts of roadblocks did you overcome? What sorts of roadblocks seemed to currently elude you, or perhaps you did not overcome to your satisfaction? Reflect on if it would be worthwhile to ever revisit these roadblocks, or reflect on how these roadblocks may have actually been formative experiences for you.
- □ Your intentions for the future (whether this involves computer science or not)
- □ Things that you wish we had time to try, or your planned future projects
- □ Any ideas for the future of the course? Was there anything you strongly disliked or strongly enjoyed doing?
- □ Of course, this list is not exhaustive, merely the topics I definitely want you to reflect on, feel free to reflect away on more topics as you wish! If you write it, I will appreciate reading it!

Appendix G

Helpful Video Links for Computer Science Teaching

Video to Assist in Modelling How to Help Your Peers

Video that may illustrate how to help peers. Remember, students notice how you provide help—you will be the number one factor in modelling how to help others, but it does pay off to also address this topic with your class! This video may assist you:

Search YouTube for: **"Medieval helpdesk with English Subtitles"**

Video URL (as of Oct 15, 2017): https://www.youtube.com/watch?time_continue=8&v=pQHX-SjgQvQ

This video may help since it shows a very patient helper helping his fellow peer—with a bit of comedy thrown in.

Video about Lifting Code from the Net (I.E. Code Plagiarism):

This can be an entire topic all to itself: I do recommend that you talk about this with your class once you get to your first major project. This is something that students will be challenged with all the time. Where you stand on the issue is up to you! I found this video to inspire some in class dialogue that is useful. **BE WARNED! There is foul language (swearing) at the 4:41 mark, the 5:01 mark, and the 5:35 mark!** You may wish to just jump to the 6:03 mark to wrap up, or summarize his ideas for your class at this point, or stop the video there!

Search YouTube for: **"Computer Science Students Should Learn to Cheat, Not be Punished for it"**

Video URL (as of Oct 15, 2017): https://www.youtube.com/watch?v=WW6Jycaol4Q

Inspirational Videos About Computer Science

These are great to show in the first week or two of class, they may help you connect the course to the real world a bit by showing students how coding can really impact human lives and our quality of living, and then relates it back to coding education in school.

---

1. Search YouTube for: **"What Most Schools Don't Teach"**
   Video URL (as of Oct 15, 2017): https://www.youtube.com/watch?v=nKIu9yen5nc

This video was created by Code.org, and follows a special initiative in the USA right now that is trying to spread awareness about computer coding. This video has a lot of big names in it—Bill Gates, Mark Zuckerberg, will.i.am, and a bunch of people from companies like FaceBook, DropBox, Valve, and others. This is a good video to show in the week of class to get students excited, but also thinking about how computer science is important in the world at large, and so central to our current life styles.

2. Search YouTube for: **"Full Length CBC Documentary: Code Kids"**
   Video URL (as of Oct 15, 2017): https://www.youtube.com/watch?v=6JGy8zmskbM

This video is about the journey towards better coding in Canadian schools. The makers travel to Estonia and Finland to investigate how they have been successful. This video really focuses on the Canadian Maritimes provinces, and has a lot of Canadian success stories in it. Great to inspire your students to code.

---

3. Search YouTube for: **"Kids Learning to Code: How England is pushing to teach coding in school"**
   Video URL (as of Oct 15, 2017): https://www.youtube.com/watch?v=YqHUKsGqUeo

This video may be more for you as a teacher, but is an interesting look at how other countries are leveraging code in schools. May be something that could help you pitch the idea to potential school leaders in order to convince them of the value of computer science in school.

Created by Matteo Di Muro
dimuro.matteo001@gmail.com

Appendix H

General Exercise Grading Rubric

Category	5	4	3	2	1	0
Coding Style & Comments	-Name of variables are intuitive and consistent -Names of Objects are intuitive and consistent with coding conventions -Code is formatted perfectly	First two bullets in Level 5, but code formatting may not be ideal	-variable names make sense -code is generally laid out nicely, easy to read, good indenting			-poor variable names -Poor indentin g of code
Program Execution & Test Runs	-The program works without any bugs or errors. -Test run values perform as expected	-One small bug or error in program execution OR -one test run value is not as expected	One small bug or error in program execution AND One test run value is not as expected	Three bugs or errors that may effect program execution extensively		Major bugs or errors in code that program use untenabl e and/or fails all test runs
OutPut Format	Everything outputs correctly and neatly in the design view of the running program					Output is incorrect or not displaye d in design view while program is running

Appendix I

A Few Samples of Possible Projects for Students in Grade 11/12 Who Don't Want to Plan Their Own Yet

THE THING SIMULATOR

You will create a text-based game loosely based on the 1982 movie "The Thing". From Wikipedia: ***The Thing*** (also known as ***John Carpenter's The Thing***) is a 1982 American science-fiction horror film directed by John Carpenter, written by Bill Lancaster, and starring Kurt Russell. The film's title refers to its primary antagonist: a parasitic extraterrestrial lifeform that assimilates other organisms and in turn imitates them. The Thing infiltrates an Antarctic research station, taking the appearance of the researchers that it absorbs, and paranoia develops within the group.

Using this premise, create a game where players must figure out which "characters" are infected… each night, there should be a chance that a new character will become infected (perhaps 25% to 50%, up to you!) There should be 10 characters, with only 1 being initially infected (should be randomized). Once 5 of the characters are infected, the game should end. Each of the six characters should have some distinctive feature, as the game will print one clue every night as to who the assimilated crew members are.

You will need to store which characters are infected in an array (watch a video here https://www.youtube.com/watch?v=c9TaEQr7uIg). You may use arrays for other things as well.

You will need to plan your game out on paper a bit (how will it work? Who will the characters be, what are the clues for each one?).

Here is some sample output from a working Thing Simulator Game:

Screen 1:

```
Welcome to The Thing Simulator!
The objective of this game is to figrue out who is assimilated
by administering tests to each of the twelve characters.
Every night, strange events will occur.
It will be up to you to deduce who is not human based off these
events. You must eliminate all assimilated hosts before they equal the survivors and swarm them...
You are stuck in a research base in the heart of the antarctic. Radios are useless due to the storm outside.
The creature aims to escape to civilization and assimilate all of humanity. You must stop it!
Press Enter to start the simulator...
|
```

Screen 2:

```
Options

Day 1

1: Blair  (Scalpel)
2: Nauls  (Radio)
3: Palmer (Switchblade)
4: Childs (Flamethrower)
5: Dr. Copper (Scalpel)
6: Norris (Flamethrower)
7: Bennings (Pistol)
8: Clark (Switchblade)
9: Garry (Pistol)
10: Fuchs (Scalpel)
11: Windows (Radio)
12: John (Pistol)


Something was set on fire last night...


Choose who to test: 4
He was assimilated! Thankfully you promptly incinerated him!
Press a number and enter to sleep for the night...
```

As you can see, the clue points to either Palmer or Clark, the only two characters who have switchblades. Your characters may have different functions or different items that could uniquely identify them as being infected. You obviously will require that some of the characters have the same function/items. You could make your game so that each character has two items (thus making it a little harder to decide who is infected).

Another example of Screen 2:

```
Options

Day 1

1: Blair  (Scalpel)
2: Nauls  (Radio)
3: Palmer (Switchblade)
4: Childs (Flamethrower)
5: Dr. Copper (Scalpel)
6: Norris (Flamethrower)
7: Bennings (Pistol)
8: Clark (Switchblade)
9: Garry (Pistol)
10: Fuchs (Scalpel)
11: Windows (Radio)
12: John (Pistol)


Someone tried to contact civilization last night...


Choose who to test: 12
He is human!
Press a number and enter to sleep for the night...
```

I am a big fan of Text Based Adventure Games—because from a programming point of view, they can literally encompass many of the most important basic concepts that are crucial in programming, and allow students to flex their creative muscles by designing intriguing stories and quests. Besides being cross-curricular, they also help students appreciate just how much work goes into creating a game or project.

TEXT BASED ADVENTURE GAME

The object of the game is to slay the dragon (or whale, or whatever other "villain" you choose, if you wish ☺). Feel free to modify the objective of your game as you see fit, this is merely an example so we can talk about the structure of the game.

The player should be given the opportunity to "beef-up" their character before facing the dragon, through the addition of side-quests (or some other game mechanic as you see fit that makes sense to your game). You will have 3 side quests. Each side quest should give the opportunity for the player to "die" (the game would be no fun without some risk!) and/or gain a few items that may help them defeat the villain. This could include health potions, speed potions, extra weapons or abilities, money to spend in the store, etc. The quests themselves should include some fighting (or some other interesting game mechanic as you see fit)!

<u>Create a Player class. The player constructor should include:</u>
Name, Health (an integer), DPH (Damage per Hit), Money
Armour (this is a value that can be deducted from incoming attacks)
Inventory (an array of size 10—the player will only be allowed ten items at once!)
Each item will need an accessor method and a modifier method!!!

<u>A Monster Class should also be created to store info about monsters:</u>
Name, Health, DPH, Armour,

<u>There should also be a weapon constructor, which allows you to build new weapons that the player can acquire:</u>
Name, Damage (this will be the amount of damage the weapon does—and hence the player does)

<u>There should be an Item class as well, for potions and other things that the player can have in their inventory:</u>
Name
Health Modifier
DPH Modifier
Armour Modifier

Depending on what the item should do, you can make the modifiers equal to 0 (for nothing) or another value if they are to do something.

You can use any other constructors that you see fit to use! Make sure they are all properly documented!

The game will make use of the clearScreen procedure we used a while back to clear the screen after each action. Your side Quests can each be a different procedure such as “sideQuest1” etc. You will use a Boolean value called “EndGame” to determine when to end the game (in case of a player death, for example, or zero health, etc.)

Starting the Game

Get the hero’s name from the user.

Every screen should always display the hero’s Name, and their vital stats at the top of the screen (look in the player constructor)

*The initial game ***could*** start like this (this is only an example, not what you have to do!):*

Welcome to <insert villain name here> Slayer! Select your action from the list below:

1) *Go on Quest*
2) *Buy Items from Hero Store*
3) *Fight the <insert villain name here>!*

If the user selects 1), then a quest that you have pre-programmed should be selected randomly. If a quest has already been completed, make sure it can’t be done twice!

Each Quest should include select case statements or if statements to give the player some choices. For example, I may have a quest that allows the player to slay a troll. It may look like this:

“You begin to walk into the woods. The air around you is thick with the sounds of strange animals. You make your way into the forest. The trees around you leer over you. You trudge on, and come to a fork in the path. You can go LEFT or RIGHT.”

```
playerInput = Next.String();
playerInput.toUpper();

Select case playerInput
    Case LEFT
            System.out.Println(“You decide to turn left into the woods. You don’t notice the
            suspicious branches and fall into a giant hole and die!”);
            endgame = TRUE
    Case RIGHT
            System.out.Println( “You decide to turn right into the woods. You find 10 gold,
            and you see a little girl crying. Do you HELP or IGNORE the girl?”);
            playerInput = Next.String();
            playerInput.toUpper();
            Select Case playerInput
                    Case HELP
                            System.out.Println(“You approach the little girl and etc. …”);

                    Case IGNORE
```

```
System.out.Println(You decide the little girl is suspicious so you
continue further down the path etc. etc. …");
```

Etc. etc. You can decide how complicated/long you would like to make your quests. Just make sure to give your player some flexibility, and rewards for actions like gold or potions or new weapons, etc. I'd say a basic quest should include 3 break points (like turning Left or Right, or 3 different actions) before it is long enough to be considered a "quest".

Combat

Combat should be turn based. To make it harder, you may want the monsters to be allowed to attack first. Players and monsters will be able to do a range of damage… i.e. a player may be able to do between 1 – 6 damage, this could be generated by a random number between 1 and 6.

You'll probably want to make a combat procedure, such as

Fight(Monster, Player)

Which will handle the fight with a do … while loop. If player health or the monster health goes below zero, the fight is over! After a fight, a player's health should be restored to its maximum with a message like "you defeat the foe, and you feel better already! Your health quickly regenerates for no reason!"

The dragon fight may be more scripted (hard-coded) if you want to make it more exciting, you may give the player decisions at certain points in the fight, or, you may just have a straight forward fight, it's all up to you! The sky is the limit!!

Take a good look at how we made the Circle Class, you'll want to make your game program in a similar way!

Assessment

Here is a rubric of how I will mark the project: keep it in mind as you are doing your game. Remember, you can make your game as complicated or simple as you wish. I recommend starting as basic as possible, and adding elements and complexity AFTER you have a basic game working. SAVE YOUR WORK OFTEN. If you make radical changes, make it in a new project!!

	0-1	2	3	4	5
Classes	ne classes are missing, or incomplete		Classes are mostly complete, with all the required elements such as accessor and modifier methods		All required classes are included, with full documentation and all accessor and modifier methods as necessary
Combat	ombat is non-existent		Combat works, but has problems		Combat works, is well explained via messages to

	or not working		(perhaps health does not display to the screen!)		the screen, and all vital stats are kept track of on the screen and it is clear what is happening
Quests	Failed to include the inimum 3 quests with 3 branches each		The minimum 3 quests with 3 branches are included, but aren't overly inspired or special		Extra effort has been taken to make sure the quests are varied, and interesting
ocumentati on	o commenting or very little commenting of code		There is documentation here and there,		Classes are documented properly, and code is documented where necessary so everything is clear
Complexity	Project is very bare ones, or components are missing, not legant, no extra effort apparent in the work		Some effort has been put into making the project creative, or extra elements are included		The project is very creative, thoughtful, and it is apparent the student has taken pride in their work and made something exceptional

Created by Matteo Di Muro

Feel free to use/modify as long as you credit me!

dimuro.matteo001@gmail.com

Appendix J
List of Some Prior Student Projects

-Complete class set of notes to learn the basic programming principles of Visual C#
-A "game theory" unit for Microsoft Visual Basic, including how to create your own sprites and eventually create a space invaders type game
-A program that was basically a very rudimentary A.I. system that was created to play a specific online game by itself.
-Various Scratch games.
-A Google Chrome extension that worked with GradeLogic (a program used by BSD teachers to enter and record grades) to send a text message to a student's phone whenever a teacher updated or created a new grade.
-A biology simulation where players had to control levels of compounds living creatures on Earth need, such as sunlight, rain, CO_2, oxygen, etc. to make populations of certain species grow or decline to reach certain goals.
-A trivia game to help students study for Grade 11 Biology, broken down by topics.
-A wide variety of games, from maze games, to click and point games, etc.
-Learn how to code in Python.
-Learn how to code in C++.

Created by Matteo Di Muro
dimuro.matteo001@gmail.com

Chapter 14 - Providing Welcoming and Engaging Environments: Making High School Computer Science Exciting and Accessible

By Kate Nizio
Computer Science Teacher
Fort Richmond Collegiate, Winnipeg, Canada

❧ ❧ ❧

"Computer Science teacher," they remark, "you must be really good at fixing computers. I've been having this issue with mine where…"

...and the new acquaintance continues to explain their issue, assuming that computer science is the study of virus removal, installing a new operating system, or maybe even just how to use an email client.

"Ah, see, no… I don't really fix computers, per se. I mean, I spend enough time on one that I might be able to help you, but as a computer science teacher, I teach kids to code. So, for example, you know that program you're having trouble with right now?"

The acquaintance nods.

"Well, someone with a computer science background probably wrote it. That issue you're having: it might be one of the programmer's little mistakes. It's what we call a 'bug'."

The acquaintance looks at you, blankly. "Hmmph. Interesting. How'd you get into teaching that?!"

❧ ❧ ❧

The Challenges

Being a computer science teacher comes with a list of challenges. The above example begins to highlight one of the overarching issues in building a successful computer science program - many people really have no idea what computer science is. In turn, how will teenaged students have an accurate idea of what it is when they're signing up for in their electives every year? It is because of this very issue that computer science teachers need to slow down and look at the underlying issues in computer science.

In order to fully understand how to create a successful high school computer science program, it is important to look at some of issues that come along with trying to engage students in the subject. The first of these challenges is the existing stereotype of a computer science student - rather 'nerdy,' antisocial, and male. Part of that picture has been proven to be correct – women account for only about 30% of graduates from math and computer science programs. (Government of Canada, Hango, 2015) From my own personal experience, I can attest to this fact. I was one of a handful of females in every high school and university computer science class I attended. As for the 'nerdy' and 'antisocial' part - for many students, those descriptions are also somewhat accurate from what I have encountered. A favorite hobby of many of the students I went to school with, and some of those whom I teach, is playing video or computer games. Their dreams of one day developing a video game themselves was the driving force behind their degree choice. The sheer nature of gaming is built on interaction between the player and a gaming console. Any interaction with other humans is online through a screen or a microphone, and in the context of a video game.

Given the fact that I am a female, rather social, and was never really into video games, how *did I* get into computer science? I always enjoyed working with computers. As a teen, my parents had to beg me to shut the computer off and go to bed instead of chatting with friends on MSN messenger. What they weren't aware of is that between sending messages back and forth with friends, I would be looking up how to modify webpages, or how to crack passwords on computers. None of this was malicious - I just thought it seemed really neat. For some reason, I thought the possibility of figuring out a way to make a computer do what I had requested through a series of seemingly secret codes was really satisfying: like cool hacker stuff.

From my experience, many students come into a computer science classroom with that general curiosity about computers and how they work. Often, students who sign up to take computer science as an elective in high school have a schedule full of other computer-related electives. In many cases, these students don't have a clue of what computer science entails - they're just excited to be able to sit in front of a computer and mess around for another period of the day.

That still does not really broaden the types of students that might find the course interesting. We now have a class of gamers and those specifically interested in computers, which does not create much diversity. What about the math-inclined female who isn't aware of the relationship between math and computer science? How about the artistic male that has the potential to design beautiful software someday, but doesn't realize the possibility of infusing programming with art? Computer science still tends to attract a very specific group, due to what the general public thinks (or doesn't know) of it.

The second problem that arises comes from the inconsistency of exposure to computer science before high school, creating a wide range of abilities within the classroom. As mentioned before, some students will have dabbled in coding through the wide array of resources now available online. Many companies are doing their best to create free and fun online tutorials or courses, as they know that the students of today will be tomorrow's employees. Some teachers in early and middle years classrooms are starting to embrace these resources as well, seeing the potential for developing critical thinking skills, and the connections to other subjects. Initiatives like *Hour of Code* are becoming more popular, and students are starting to see code through explorations in newly developed makerspaces. Although this is a fabulous way of introducing programming to a greater audience than the ones earlier discussed, the initiatives are currently not reaching all students. This creates classrooms where some students have written various programs, where others have not written a line of code in their lives.

At this point, you can start to see a divide that may create another point of difficulty in teaching computer science. Attracting a student to take the course is half the battle. The other half begins on the first day when a student actually enters the classroom. This is where things get tricky: students may take the course for many of the reasons listed above, but will the course live up to their expectations? The idea of creating the game of your dreams, messing around with computers at your own pace, and trying an *Hour of Code* IS different from a computer science course.

The truth is: for some students, computer science can be a really challenging subject.

This is why it is incredibly important to introduce students to coding who know nothing about the topic without overwhelming them. As earlier mentioned, some students may be familiar with computer science through online tutorials, while others may have never seen code in their life. To illustrate, one student may have independently completed *Khan Academy*'s Javascript tutorials, and because of it, they breeze through your introduction assignments. Unfortunately, the student beside them has never seen code in their life is still struggling with question #1 while the other student completes the assignment. Overwhelmed by syntax like *int, Math.random(), public static void main,* et cetera, the struggling student can't even gain essential concepts, such as the fact that computers read code sequentially, or that a variable is similar to variables in their math class. Trying to teach computer science in a traditional language, like Java or C++, creates a lot of instances where you have to say "don't worry about those words, we'll cover that much further down the road." The sheer power of these languages make them a tough place to start teaching students how to code.

On the other end of the spectrum, if students aren't feeling particularly overwhelmed, some may end up feeling incredibly bored. The dry, mathematical nature of beginner problems in traditional languages can be a real turn off for students who are not mathematically inclined. Furthermore, it is difficult for students to see the correlation between programming a function that codes Fibonacci's sequence and building a video game, leaving some students feeling rather disconnected from the reason they took the course in the first place.

Solutions

Despite the various challenges of teaching computer science at a high school level, there are ideas that I believe can remedy the problems mentioned above. The solution I propose is not particularly piecemeal, but deals with these issues holistically by using the ideas of James Paul Gee in conjunction with a variety of new resources and development environments made readily available to teachers.

Gee's tanks and sandboxes

In order to outline some solutions for the various problems earlier mentioned, it is important to know a bit about Gee and the ideas in his book *Situated Learning*, specifically the chapter titled *Learning and Gaming*. Ironically, the idea that may help students who are not "gamers" to become engaged in a topic like computer science is illustrated by Gee through a video game tutorial analogy. (Aside: Although this sounds typical of a computer science teacher, let me assure you, I am not a "gamer." If you were to have me play anything beyond Super Mario 3 for the old Nintendo Entertainment System, I would flounder. I can only handle two dimensions. Once a third is involved, it is "game over" for me.)

When speaking about his ideas of the *Fish Tank, Supervised Sandboxes, Unsupervised Sandboxes,* and *Learning and Playing*, Gee makes some very interesting points about a video game's ability to teach:

> "First, learners are all different and the designers [of video games] don't know what each one already knows, nor what their favored style of learning will be. Second, learners don't necessarily know themselves how much they do or do not already know and what their best style of learning will be in a given situation. Schools tend to handle these problems by assessing the learner and then deciding for the learner how these problems ought to be dealt with. [Many good games] solve the problem by letting learners assess themselves and learn things about what they do and do not know and what style of learning suits them here and now. Learners then decide for themselves how they want to proceed." (Gee, 2009, 56)

I propose that there is a really fabulous connection between his analogy and teaching computer science specifically. There's something about coding that lends its hand to Gee's theory that is rooted in the nature of a computer program. In computer programs and in gaming, it is the computer – this inanimate object - that tells you if you are right or wrong. Students are not at the mercy of grading by a teacher. When the compile button is pressed, a program is run, or a game is played, a student gets immediate results from a non-judging machine that says "you're right" or "you're wrong." The difference here is that a computer gives you the option to try again without fear of human judgment. Although it is our responsibility as educators to provide formative feedback and to try to ensure there are no gaps or deficiencies in student knowledge and understanding, running code inherently provides feedback for students.

Additionally, simply having technology, such as a computer, allows for so many more options in delivering knowledge, and also allows for students to have a little bit more choice in when and how they require support in their learning. By including accessible resources, similar to the ones available in game tutorials, learners can be empowered to learn when and how it suits them. This idea, combined with many of the new development environments, makes it easy for students to take control of their learning, much like the learning that happens in a video game tutorial. It is in this way that we can encourage students to take charge of their own learning, and to see the potential for success in a field such as computer science.

Development Environments

The second part of the solution depends on the vast number of development environments available that help students not only learn code, but to do it in a way that isn't as math based as traditional computer science education. These resources provide opportunities for more authentic learning experiences, where students are motivated by seeing something cool happen in a game or program. Learning specific concepts can often be framed in a way that seems more like a game challenge on a computer, versus learning a concept for the sake of a grade or passing a course. As Gee mentions of video game tutorials, "Information is always given "just in time" when it can be used and we can see its meaning in terms of effects and actions." (Gee, 2009, 60) These development environments provide ways to create exciting effects that motivate students, giving teachers the opportunity to provide information on how to do something cool right when the students need it.

For each of these environments, I often have various tutorials and challenges that I provide in a variety of formats, including modeling the process of programming and problem solving in front of the whole class, online slides, YouTube videos, and self-guided tutorials. Many of the resources for each project come in various forms to provide support when necessary, which will be discussed later on.

Beginner (fish bowl): Block-based programming languages

Block-based programming languages give students the ability to learn general computer science concepts without the annoyances of traditional code. Students can actually learn about general programming concepts, such as variables, loops, and conditionals, in a way that allows them to focus on computational thinking instead of syntax.

One program that does this very well and has many free resources available online is Scratch. (https://scratch.mit.edu) Although scratch is marketed towards a younger audience, it can be used to make relatively complex games. I often start with a full class tutorial that helps them to learn the basics of Scratch, and then have them move to more comprehensive, self-guided tutorials that allow them to move at their own speed. Luckily, Scratch has many tutorials and references built right into their online interface. For students who have coding experience that race through the basic tutorials, I often guide them to try a tutorial of their choice, such as ones that teach them how to jump, shoot, bounce, or have scrolling backgrounds. These advanced tutorials end up providing great amounts of background knowledge that can be transferred into games built with more complex code later on. Although these skills are not specifically requested of students in their project at the end of this unit, the assessment is open enough that students can show off these techniques, or stick to the basic requirements I outline for the end of the unit game that students create. The excitement of creating a more complex and exciting game is often enough to encourage students to learn something beyond the basics of Scratch.

Intermediate (pond): Simplified traditional languages

The step into actually typing out code can be exciting for some, yet incredibly daunting for others. Although the program I teach leads up to an option to write the AP Computer Science A exam in Grade 12, I try to stay away from teaching any object oriented programming concepts until students have had the exposure to syntax and how very precise they have to be when coding. Therefore, I like to step from block-based languages to working with a simplified version of a traditional language. Choosing something with less of a learning curve than full-blown Java gives students the chance to practice what they've just learned about how to build programs without becoming too frustrated.

I find Processing (https://processing.org/) to be the perfect "in between" language. Much like what it was originally created for, it teaches "programming fundamentals within a visual context." (Processing Foundation) Processing is built on Java, which means that the syntax for method signatures, calling methods, creating and using variables, and coding conditionals is very similar, if not exactly the same, as Java. Not only is the language simplified, but it also allows beginner coders to create something tangible in very little time. This has made it a favorite for visual designers and artists, which also holds natural appeal for students who are not as math-inclined. The motivation to make something exciting tends to help propel students through the frustration of missing a semicolon or bracket.

In my classroom, students start off by creating their choice of bears or robots using methods that draw shapes and colours. Then, they're challenged to start to make shapes follow the mouse or move across the screen in certain directions. Soon, they know enough to create a virtual magic-8 ball with their own personal phrases, or create a tortoise and hare racing game. Before students know it, they know how to call a method, create a float or boolean variable, add PImage objects, make things happen when buttons are pressed or the mouse is clicked, and understand vocabulary like "parameter" or "method signature." Of course, basic Java syntax is also learned, such as the fact that lines of code end with a semicolon, and how to start and end a conditional block of code with curly brackets. It should be mentioned that Processing can be used to teach more advanced concepts, such as arrays or objects, but can also be used in a basic way that doesn't require a main method and loads of extra brackets or words that students may become confused about.

Additionally, the creative nature of Processing allows for open-ended assessments. In the final Processing assignment that I assign to my students, they are encouraged to create some sort of scene that shows a range of skills and understandings. It has to be something somewhat real – not so much abstract – so that a student can clearly outline their goals in the scene. Other than that, students are encouraged to be as creative as possible. This is a stark contrast from strictly prescribed problems and tests, which tend to be a staple for traditional computer science programs.

Advanced (lake): Visual Object Oriented Programming

After dabbling in Processing for around a month's time to become a little more comfortable with the demands of coding syntax, I like to make students feel a little uncomfortable again. This time, we take a step into an IDE called Greenfoot (https://www.greenfoot.org/door), which allows students to learn about Object Oriented programming in Java. Greenfoot is great because it provides a really easy way to see new instances of objects, and how two objects can be the same type, yet have different states. Students learn to create classes that extend from two main classes: Actor and World. Actors can be added into the world, and thus, you can see how this environment fosters game creation. Right from the beginning, students have to learn about classes and how to inherit methods, which is a concept not usually learned about until part way through a university degree.

This very fact often blows the mind of traditionalists, who think that the concept of objects is too much for new programmers. In a way, they are right – the idea of objects is so abstract, and in many ways, very complex. With an environment like Greenfoot, students begin to use objects without fully grasping the concept of them. While this sounds counter productive, Greenfoot gives students the opportunity to really *experience* objects and start to build an understanding of them, versus only being introduced in very abstract terms sometime down the road. Regardless, to create a basic game, you need to create classes that extend the actor and world classes. Even if students are unaware of what this really means, a greater understanding builds over time through use, much like the understanding built in the tutorial of a video game.

Although some students are unaware of the possibilities within Java and object oriented programming, many of the students who are particularly talented in the area of computer science can take off with the ideas of objects. While novice programmers are replicating examples where you've called built-in methods, more apt students are starting to see how returning methods work, and how to call methods from other objects. I will often teach the idea of documentation and how to use it rather early in the exploration of Greenfoot specifically for the reason that once advanced students become a little more comfortable, they are able to find and use methods that they may not have directly learned about before.

As mentioned, Greenfoot lends itself to being perfect for game creation. With methods that check to see if two actors are touching, ones that allow you to easily move and turn the actor object, and others that allow you to set actors' images, most tutorials and assessments focus on making games. After seeing how they can use the available methods to create basic games, students are then given the opportunity to create their own game. Depending on how far we make it in the course, the student's final project is to create a shooter-type game. Students who are not as comfortable with coding usually stick to making a one-way shooter game, while other students make make a multi-way shooting game. For those students who zoomed through the basic tutorials, they are once again encouraged to build upon their basic knowledge by creating games with scrolling backgrounds and jumping. In some cases, these students already gained the concepts of these games through Scratch, and are now able to take what they learned then and implement such games using real Java syntax.

Taking it outside the computer (the landscape): Exploring other areas of computer science

Although I do focus my introductory computer science course on game creation and moving towards learning object oriented programming, I like to add a small unit that introduces students to another area of computer science. This provides another chance for students to connect to the subject, and opens their eyes up to other career options in the field.

The options for what to touch on outside of general programming skills are vast. One that I've been using in my classroom is the Lego NXT/EV3 robots (https://www.lego.com/en-us/mindstorms). These give students a general idea of how a robot might deal with information coming in from the outside world. From following lines to completing challenges that require the robot to pick up items and put them down, the Lego bots give students a neat little introduction to robotics without having to build one from absolutely nothing.

In the case that you do want to have students create robots from nothing, Arduinos (https://www.arduino.cc/) can be very useful. The simple microcontroller gives students the opportunity to make a wide array of interactive projects, while introducing them to circuit boards and wiring. I am only just unpacking the opportunities of Arduinos, but have seen many unbelievable projects along the way. One of the best parts of using Arduinos is that it is built on Processing, and uses a very similar development environment.

One of the biggest reasons to learn to code these days has to do with app development for mobile devices. While most app development software comes with a massive learning curve, App

Inventor 2 from MIT (http://appinventor.mit.edu/explore/) allows students to use a block-based programming language to build apps that they can test directly on their phone. Creating apps that can actually be tested on their phone is not only a novelty, but also introduces students to event-driven programming.

Putting It All Together

With the ideas of Gee and new development environments available in hand, it is now up to the educator to structure a program that allows for students to help themselves advance in ways that they see fit, and to give them the space to produce projects that interest them. While the new visual and interactive environments create new ways for students to connect with computer science that may not have typically gravitated towards it in the first place, it is up to teachers to provide resources and supports at times when students need them the most.

This “need for support” idea exists in two ways, both dealing with the different levels of understanding that students may be coming with into a computer science classroom. The first way would be the traditional need for support – when a student is struggling with the subject. Like many teachers, I provide a few example problems through a whole class tutorial, but often, students require a re-teaching or restating of concepts. Given that we work in a computer lab, the technology available to us allows for not only posting notes and other examples, but also allows us to create screen captured lessons, as well as other helpful links to resources on the internet. This, combined with frequent conferencing and checking in with struggling students makes the subject that much more attainable.

The other “need for support” exists with students who find the subject to be easy. These are the students who really could do without a full class tutorial, and could probably figure things out just from a teacher providing some basic written notes. With these particular students, it now becomes essential to provide them with new challenges and project ideas. I personally have a “turbo coders” section on my site with many opportunities for extending knowledge and chances to problem solve. Again, with the power of the internet, there are many pre-existing resources available to keep students engaged and learning. Whether through these visual programming languages, or by simply by mucking around in the development environment, gifted computer science students can use these resources to make games or robots beyond the abilities of their classmates. By keeping summative projects open and flexible, this also provides an incentive for students to extend their understanding beyond the basic outcomes.

Conclusion

Despite the challenges of teaching high school computer science, there are many ways of improving what has been the status quo in computer science education.

1) Providing students with various environments and opportunities to encounter code, we can begin to engage and interest more than just the "typical" computer science student.
2) Beginning with safe, easily accessible languages and slowly introducing more complex ones, we provide fish tanks in which students can become comfortable, and then lakes and worlds which they can then advance to.
3) As languages become more difficult, it is important that our tutorials come in various forms of resources, including spoken lecture, simplified text, and videos, so that we can provide support for a wide range of abilities. Keeping assessments open ended and creative, and providing resources for enrichment, students who excel at computer science can still be challenged.
4) Given the visual and interactive nature of newer development environments, the wonder and excitement of making cool things happen on the screen provide opportunities to teach theoretical concepts when they matter to the student most. This creates a much more authentic learning experience for a broader audience.

Through these ideas, we may be able to start to involve females and other less-represented demographics to computer science. As a result, this may bring about new perspectives and ideas into the world of program development, creating technological solutions that represent and are understood by the greater population. Even if these students do not continue on in computer science, learning to code provides a fabulous opportunity to learn to think computationally and improve a student's critical thinking skills. None of this has to come at a cost – students who excel at computer science are not held back by these ideas, but are constantly encouraged to expand their understandings and be creative with their programming.

Although not truly groundbreaking, this is simply a portrait of what an interesting analogy, engaging environments, and good teaching can look like in a high school computer science classroom.

A huge "thank you" goes out to Kevin Rowan, for passing on his program to me so I could keep growing it.

References

Gee, J. P. (2009). *Situated language and learning: a critique of traditional schooling*. New York: Routledge.

Government of Canada, Hango, D. (2015, November 27). Gender differences in science, technology, engineering, mathematics and computer science (STEM) programs at university. Retrieved November 10, 2017, from https://www.statcan.gc.ca/pub/75-006-x/2013001/article/11874-eng.htm

Processing Foundation. (n.d.). Overview. A short introduction to the Processing software and projects from the community. Retrieved November 10, 2017, from https://processing.org/overview/

Conclusion

By Mike Nantais

This eBook has taken a look at the Maker and Coding scene in Manitoba schools. Are these just new trends that will fade away like so many in education have done before, or are these here to stay? Only time will reveal the answer, yet these initiatives have been taking education by storm and, here in Manitoba, we can see many powerful examples, a sample of which are provided in these pages. In the introduction to this volume, Rennie Redekopp writes, “these are not the ‘silver bullets’ that will fix education. They are great ways to think about how we teach and how students learn, and must be thoughtfully implemented only after we decide what is really important in the lives of our students.” We have read in this book how these ideas are being approached in Manitoba schools. We now have been provided with excellent resources and powerful ideas for engaging students in learning through Making and Coding. The two trends that form the subject of this book also have some overlap, often coding is included as part of Making, for example, in robotics or with “makey makey”, however, making is not just about technology. So, what can we learn from the chapters and stories in this book?

Looking back, we realize that these ideas are not new in education. Both go back decades, however, they are experiencing a new life. As described by Shauna Cornwell in chapter 2, “Spaces where students make and create are nothing new to schools”, citing activities such as art, dance, industrial arts, and others. David Gamble reiterates this fact in chapter 8 when he looks at connecting Industrial Arts with other subject areas in the high school. In a similar vein, computer science classes have been offered in many schools for decades. In fact, my own journey in computer science and coding goes back to the early 1970’s when I took computer science courses in high school - anyone recall punch cards and mainframes (see figure 1)?

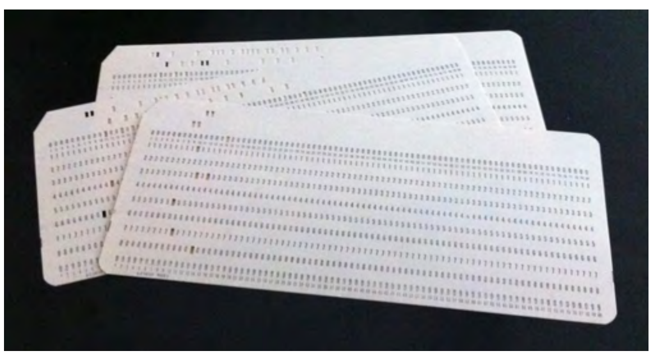


Figure 1. Blast from the past: computer punch cards
(photo by M. Nantais)

The first half of this book is all about Making, and more importantly, as several of our authors point out, the Making mindset or culture of making. Many schools in Manitoba are delving into deep learning, based on the C’s, which includes creative thinking, a cornerstone of Making.

Another important aspect of Making is the idea of active learning, with its roots in learning theories of constructivism and constructionism. These theories provide us with evidence and a theoretical underpinning for the value of activity, play, and social learning. In this section of the book, we are provided with numerous resources and ideas for implementing Making, but more importantly, we learn about the mindset necessary for successfully implementing Making. It requires giving students control, allowing and becoming comfortable with student choice, encouraging learning from mistakes, and exploration. We see from the stories provided that making helps to build resilience, collaboration, curiosity, and problem solving skills.

Our authors take us from makerspaces in the library or another central location in a school, to taking Making into the classroom as part of the curriculum. Examples show that the benefits of Making can be realized at any level, from the smallest of children in early years, in middle years, and for the 'big kids' in high school. One of the major points made across chapters is that the Maker culture necessitates a change or shift in the role of the teacher. The approach utilizes the strategies of inquiry and problem based learning, and allowing for student voice and choice.

The second section of the book transitions from Making to Coding, or more appropriately, computational thinking, another current hot topic in education. Computational thinking can be defined as the "thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer—human or machine—can effectively carry out" (Wing, 2014). Wing goes on to describe that computational thinking is not just a way of problem solving, but also a way of problem formulation. Coding is intimately tied to computational thinking in that it is the process of turning the solution to a problem (the algorithm) into a computer program. Although coding is a subset of computational thinking, the term is often used by some to refer to computational thinking itself. In recent years several jurisdictions are implementing computational thinking into their curriculum. These jurisdictions include the UK, and in Canada, British Columbia and Nova Scotia. In Manitoba, a working committee has been struck by Manitoba Education to start looking at computational thinking as part of the Literacy with ICT continuum from K to 12. There is a large amount of research into this area as well, a recent search for peer reviewed articles (from 2014 - 2017) using the keyword 'computational thinking' on Brandon University library databases yielded over 10 000 hits. Coding, computational thinking, computer science - all these related concepts have been around for a long time, but they are experiencing a resurgence.

The chapters in the second section of this volume explore implementing coding at all grade levels. Education student, Eleni Galatsanou Tellidis (chapter 11) provides an overview of the movement and some of the arguments for and against Coding for all. Coding in the early years is explored by Leah Obach and Devon Caldwell (chapter 9) and then we see a whole school approach from K-12 in Kirstan Osborne and Elizabeth LaPage's chapter (10). These show how coding can be incorporated in powerful ways across the curriculum. The last set of chapters look at various approaches to Computer Science in the high school. In these chapters we see a trend in computer science teaching that tries to be more inclusive and more student centered as we see the concepts being important in many fields and for more than just those traditionally interested in programming. We also see work in computer science expanding from learning to program in a particular computer language to thinking about problem solving and considering what it is we are actually creating when we program. The future of computer science in schools looks exciting!

To conclude this book, we can list several take aways. Both Making and the approach to Coding (computational thinking) encourage student voice, student choice, passion, self-direction, and the right to fail. They promote creativity and problem solving skills. Will these trends stick around? Are they buzzwords that will fade away? Once again, that remains to be seen. Whether these new buzzwords and approaches come into fashion or not, the main takeaway from this work, though, is that the ideas, and types of learning that result, are timeless and valuable when combined with other things we do in schools.