Collaboration, Analysis and Problem Solving in the Digital Technologies Classroom

Jessica Petersen | 1503729

Introduction

There is a common misconception that programming is closely related to math, and that in order to be a successful programmer, you need to be an exceptional mathematician. This is not the case (Khazan, 2015; Mei, 2014). While some skills may be transferable and related, programming is the art of logic, not of mathematics.

Software engineering is a field that requires more than just technical skills in order to be successful. Soft skills - skills that rely on interaction with others, such as communication, and collaboration - are just as important when it comes to growing as a successful software developer (Fernando Capretz, 2014).

When I first became a software engineering student, I understated the importance of everything that was not a technical skill. I believed that the most important part of developing software was coding and making everything work. I had no idea of how much of my computer science education would rely on the development of problem solving, collaboration, and analysis skills.

Some programmers argue that abstraction - the ability to see a property in isolation - the most important skill that can be transferred from experience with mathematics, but I disagree (Loop Space, 2017). While abstraction is certainly important, being able to examine things in isolation is just that - *examining*. I believe that analysis and problem solving work together as an extension to abstraction, and provide a more valuable application than merely being able to see a component.

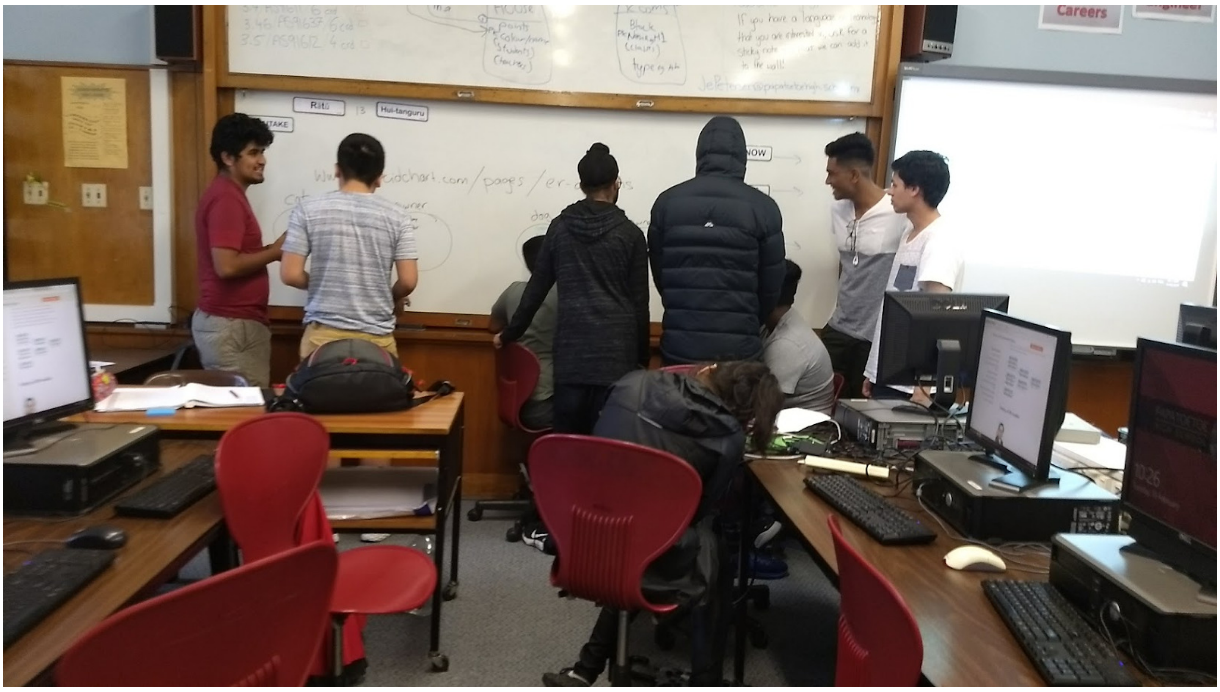
Analysis and problem solving go hand-in-hand in mathematics, just as they do in software engineering. Being able to analyse an issue in someone's code, and then take steps to solve that issue, is a key skill of a successful software engineer. It is a skill that does not only revolve around someone's ability to type code into a text editor, and a skill that is essential to growing in the programming world.

The final skill that I believe is indispensable to a successful software engineer is the ability to collaborate. While it relates to Loop Space's second most important transferable skill from mathematics to programming, it is not exactly the same. Loop Space suggests that being able to put aside your own views to make something more useful or optomised - perspective - is one of the best skills for a software engineering student to develop (2017). Again, I believe that it is far too narrow of a scope for such an important skill. Being able to approach others and work with them to produce better work overall is a much more valuable skill than accomplishing the same thing alone. Working with others gives students a chance to experience *ako* - reciprocal learning - as well as to gain the perspectives of others, from different backgrounds, cultures, and thought-processes (Ministry of Education, 2013).

Collaboration also allows for students to succeed with productive partnerships - a guiding principle of the Maori Education Strategy, *Ka Hikitia*. Working with our whanau and our peers in programming rōpū allows students to create better outcomes for all stakeholders involved in the production process (Ministry of Education, 2013).

These skills are all essential for success in my subject area. In my classroom, the development of these skills will manifest themselves through students interacting with one another, either to solve problems, ask questions, or improve their outcomes; students using diagrams and consulting documentation to improve their designs; and students debugging their software by systematically testing steps in their code independently.

In my classroom, success in developing these skills looks like this:



Above: ("DTP3 on Tuesday 13 February, working on entities and relationships", 2018).

Collaboration

The above image was taken on a day that I had planned to introduce my students to Microsoft Access - the newest tool that they would utilise in their learning. It did not work out as planned.

My students did not feel comfortable with pushing through to the next topic as they felt that they did not have a thorough understanding of entities and relationships between them. Reacting to their needs, I came up with a task on the fly - split the class into two groups. One group was tasked with creating an entity relationship diagram for the relationship between a cat and its owner, the other group was tasked with creating an entity relationship diagram for the relationship between a dog and its owner.

What I saw was interesting.

I had always tried to encourage my students collaborate through setting group work and expecting my students to find experts in each other by consulting three of their peers in regards to a question before asking me, but this was something completely different. This was *organic*. I didn't have to ask my students to collaborate - they were doing it on their own. They asked each other questions - "Why is the ID the primary key, and not the cat's name?" - they scribbled on the board, explaining their thought processes and walking through the example. My students were truly experiencing ako - learning from each other - completely without my input.

I had many fewer questions from my students about relationships between entities after that lesson. My students had collaborated and built a mutual understanding of the topic - they had worked together to achieve an improved result (Petersen, 2018).

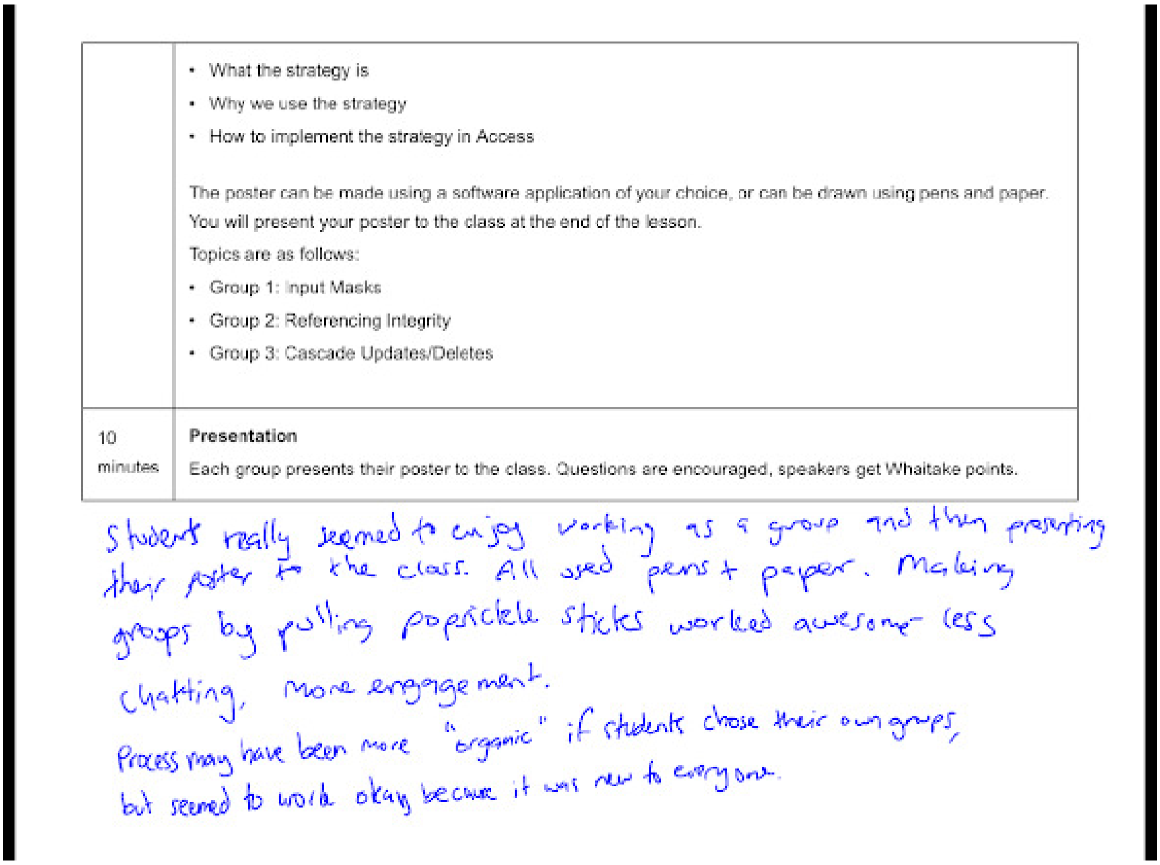
I have tried to reproduce that effect on several occasions - most effectively in late March when I planned to get my students to explore different types of data integrity preservation techniques. There were three groups, selected by random, and each group was tasked with creating a poster demonstrating what the technique was, as well as how to implement it in a Microsoft Access database.

Above: ("DTP3 on Wednesday 21 March, presenting different data integrity preservation techniques", 2018).

In that lesson, my students were collaborating in a different way. They did not have much prior knowledge around the topics, and as a result, focused on collaborating by researching as a group, and pooling their findings. This went well, and my students were able to learn about different techniques by observing the posters made by other teams during presentations.

The connections were not as natural as they had been in the previous lesson - my students were, after all, working with their classmates who were not necessarily their best friends. While it’s important to learn to collaborate with others who you may not know very well, this meant that the ako was less pronounced, and a lack of whanaungatanga with their rōpū meant that they were not as comfortable asking questions and working together.

If I did this again, I might consider allowing each student to pair up with another, and then allocate the pairs to groups.



	• What the strategy is • Why we use the strategy • How to implement the strategy in Access The poster can be made using a software application of your choice, or can be drawn using pens and paper. You will present your poster to the class at the end of the lesson. Topics are as follows: • Group 1: Input Masks • Group 2: Referencing Integrity • Group 3: Cascade Updates/Deletes
10 minutes	**Presentation** Each group presents their poster to the class. Questions are encouraged, speakers get Whaitake points.

Students really seemed to enjoy working as a group and then presenting their poster to the class. All used pens + paper. Making groups by pulling popsicle sticks worked awesome - less chatting, more engagement.

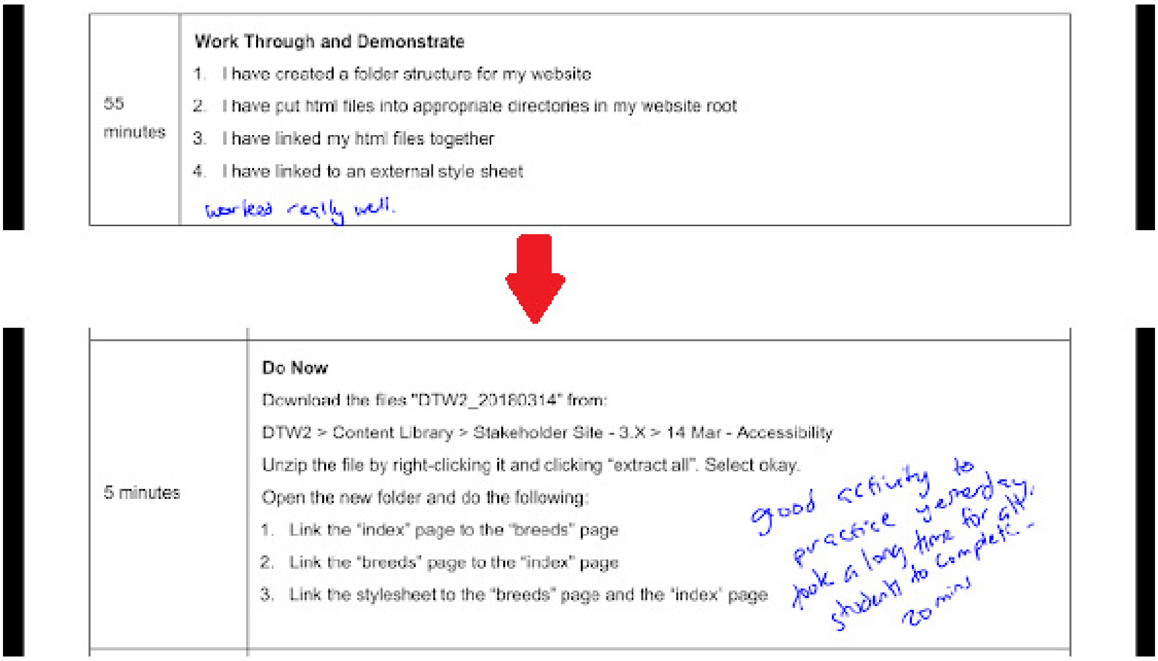
Process may have been more "organic" if students chose their own groups, but seemed to work okay because it was new to everyone.

Above: Excerpt from my lesson plan for DTP3 on Wednesday 21 March ("Databases - 3.41/page_45", 2018).

Analysis and Problem Solving

I think that in digital technologies the skills of analysis and problem solving tend to go hand-in-hand, and because of this, I'd like to address them together.

Something that I really focused on was the development of analysis and problem solving skills during formative assessments. During my initial "teaching" phase of the website development for a stakeholder unit, I was working on skill building with my students. Things that we focussed on included creating headings and paragraphs, importing images, and linking pages together. We would concentrate on building the skills and practicing them during the lesson, and then the following lesson, the do now would be a formative assessment of the skills developed in the previous lesson.



55 minutes	**Work Through and Demonstrate** 1. I have created a folder structure for my website 2. I have put html files into appropriate directories in my website root 3. I have linked my html files together 4. I have linked to an external style sheet worked really well.

5 minutes	**Do Now** Download the files "DTW2_20180314" from: DTW2 > Content Library > Stakeholder Site - 3.X > 14 Mar - Accessibility Unzip the file by right-clicking it and clicking "extract all". Select okay. Open the new folder and do the following: 1. Link the "index" page to the "breeds" page 2. Link the "breeds" page to the "index" page 3. Link the stylesheet to the "breeds" page and the "index" page good activity to practice yesterday. took a long time for all students to complete - 20 mins

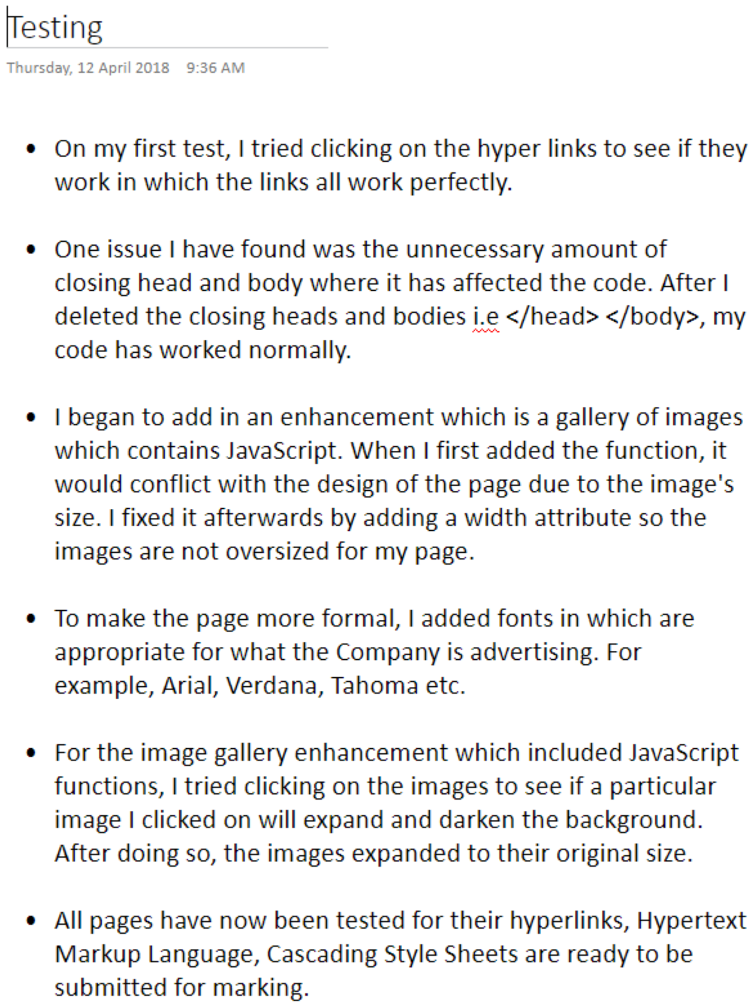
Above: Excerpt from lesson plan of DTW2 lesson on Tuesday 13 March ("Stakeholder Site - 3.X/page_46", 2018), followed by excerpt of lesson plan of DTW2 lesson on Wednesday 14 March ("Stakeholder Site - 3.X/page_47", 2018).

Students were encouraged to complete the activities by themselves with support from their classmates where necessary. In most cases, students would become “stuck” on an issue where there would be a problem with their code. Often, the issue would be the simple matter of a missing triangle bracket or misspelled word, and asking the students to read their code aloud to me helped them to solve their own problems.

Students also tended to turn to me from help instead of analysing their code by reading through it to figure out what the problem was. By encouraging them to reread their code, or directing them to resources, I allowed them to develop their analysis and problem solving skills as they’d often find the answer on their own.

This was only partially effective in helping my student to develop analysis and problem solving skills as they were not the key focus of each exercise - they were merely a by-product of the end output - the code that they had practiced. I think that by providing students with intentionally broken sample code, and asking them to fix it may be a better solution to support them in developing their skills.

Some students, however, did a fantastic job of analysing their own code and building solutions to fix this. Below is an example of this work.



Testing

Thursday, 12 April 2018 9:36 AM

- On my first test, I tried clicking on the hyper links to see if they work in which the links all work perfectly.

- One issue I have found was the unnecessary amount of closing head and body where it has affected the code. After I deleted the closing heads and bodies i.e </head> </body>, my code has worked normally.

- I began to add in an enhancement which is a gallery of images which contains JavaScript. When I first added the function, it would conflict with the design of the page due to the image's size. I fixed it afterwards by adding a width attribute so the images are not oversized for my page.

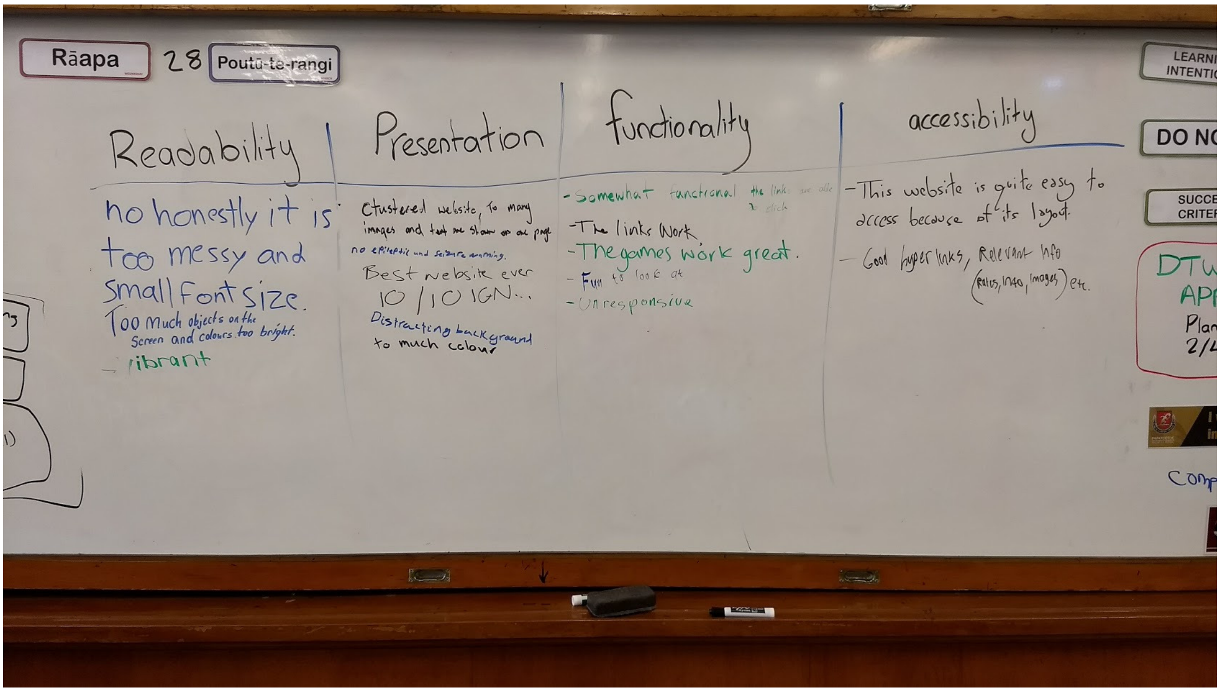
- To make the page more formal, I added fonts in which are appropriate for what the Company is advertising. For example, Arial, Verdana, Tahoma etc.

- For the image gallery enhancement which included JavaScript functions, I tried clicking on the images to see if a particular image I clicked on will expand and darken the background. After doing so, the images expanded to their original size.

- All pages have now been tested for their hyperlinks, Hypertext Markup Language, Cascading Style Sheets are ready to be submitted for marking.

Above: Student work in regards to analysing and problem solving issues in their code (Durban, 2018).

Another way that I have incorporated analysis and problem solving skills into my classroom is with the use of exemplars. One of my most recent lessons was concentrated on evaluating websites and optomising their effectiveness. The “Do Now” of that lesson was for every student to examine the “Ling’s Cars” website and to make a comment about one of the following aspects of its design: readability, presentation, functionality, and accessibility. Each student was expected to contribute their idea to a classwide brainstorm on the whiteboard.


Rāapa
28
Poutū-te-rangi
Readability
Presentation
functionality
accessibility
no honestly it is too messy and small font size.
Too much objects on the screen and colours too bright.
– vibrant
Clustered website, to many images and text are shown on one page
no epileptic and seizure warning.
Best website ever 10/10 IGN...
Distracting background to much colour
- Somewhat functional the links are able to click
- The links Work
- The games work great.
- Fun to look at
- Unresponsive
- This website is quite easy to access because of its layout.
- Good hyper links, Relevant info (Rates, Info, Images) etc.
LEARNI
INTENTI
DO NO
SUCCE
CRITER
DTW
APP
Plan
2/4
Comp

Above: ("Brainstorm of Evaluation - Ling's Cars Website", 2018).

This was great to get the students thinking about what makes a good website, and then apply that knowledge to analysing an existing website. Overall, I think this activity was fantastic in developing the analysis skills of my students, but not as useful when it came to developing their problem solving skills.

Some students identified things like “small font size” being a barrier to readability, and as a result were able to produce ways of fixing that issue. Other comments such as “unresponsive” had fewer clear courses of action to fix as the comment itself was not very specific.

In order to further improve this activity, I would ask students to take the comments that they extracted from their analysis and then come up with a plan of action to fix those issues by problem solving.

Final Thoughts

This reflection process has allowed me to gather my thoughts about what I expect collaboration, analysis, and problem solving skills to look like in my classroom. I realise now that a number of these essential skills are being covered by me in my normal teaching without special focus, but these lessons do not necessarily implement the most effective strategies to help my students to learn these skills. I need to further focus on providing activities that actively focus on skill building in the areas of collaboration, analysis, and problem solving in order to benefit my students the most. I still have a long way to go when it comes to creating the environment for this to happen, but my daily reflections of my lesson plans helps me to organise my thoughts and point out how activities could be better run next time.

References

Durban, V. (2018, April 12). DTW2 [ONE].

Fernando Capretz, L. (2014). Bringing the Human Factor to Software Engineering. IEEE Software, 31(2), 104. doi:10.1109/MS.2014.30

Khazan, O. (2015, September 02). You Don't Have to Be Good at Math to Learn to Code. Retrieved April 17, 2018, from https://www.theatlantic.com/technology/archive/2015/09/you-dont-have-to-be-good-at-math-to-learn-to-code/403342/

Loop Space [Stack Exchange User]. (2017, April 13). What does mathematics have to do with programming? Retrieved April 17, 2018, from https://softwareengineering.stackexchange.com/a/137075

Mei, S. (2014, July 15). Programming Is Not Math. Retrieved April 17, 2018, from http://www.sarahmei.com/blog/2014/07/15/programming-is-not-math/comment-page-2/

Ministry of Education (2013). Ka Hikitia: Accelerating Success 2013-2017, the Maori Education Strategy. Wellington: New Zealand Government. Retrieved January 10, 2018 from https://www.education.govt.nz/assets/Documents/Ministry/Strategies-and-policies/Ka-Hikitia/KaHikitiaAcceleratingSuccessEnglish.pdf

Petersen, J. (2018, March 28). Brainstorm of Evaluation - Ling's Cars Website [Digital image]. Retrieved April 18, 2018, from https://photos.google.com/photo/AF1QipMe1usf0F4PZDZpmBf2HXhEmebNlJ9LGF6zA94P

Petersen, J. (2018, March 21). Databases - 3.41/page_45 [PNG].

Petersen, J. (2018, February 13). DTP3 on Tuesday 13 February, working on entities and relationships [Digital image]. Retrieved April 18, 2018, from https://photos.google.com/photo/AF1QipPWIou0-VXs-kHGDDXYI6Tm5ZH7bXDIX9VA5S79

Petersen, J. (2018, March 21). DTP3 on Wednesday 21 March, presenting different data integrity preservation techniques [Digital image]. Retrieved April 18, 2018, from https://photos.google.com/photo/AF1QipNiHPgSpHKwMniJlr_GrbgUj3WINHPcidVUpnEi

Petersen, J. (2018, March 13). Stakeholder Site - 3.X/page_46 [PNG].

Petersen, J. (2018, March 14). Stakeholder Site - 3.X/page_47 [PNG].

Petersen, J. (2018, March 5). Sunburn and Burn-Out - Week Four and Five, Term One, 2018 [Web log post]. Retrieved April 18, 2018, from https://digitechdiaries.tumblr.com/post/171515933082/sunburn-and-burn-out-week-four-and-five-term