Activity Name: Parameter passing in modular code

1. What is the activity attempting to teach?

   The activity aims to allow students to gain an understanding of parameters and parameter passing techniques, and as it is just an introductory lesson, topics such as the difference between functions and procedures are not considered.

2. How long did it take to complete?

   Recommended time is 1 period (~55 minutes)

3. What should students be able to do after successfully learning with the materials?

   Students should be able to describe what a parameter is, and how it is used to create modular programs, be able to explain how pieces of data are passed into and out of a subroutine, when given a problem and a set of main steps, identify the pieces of data needed by each subroutine and whether they should be passed in, or out, or in/out, explain the difference between passing by value and passing by reference, and when given a modular program, identify when a local variable will be created and destroyed.

4. What are the learning objectives?
   - Understand the difference between a non-modular and a modular program.
   - Understand what a parameter is and how they help us to create modular programs.
   - Understand different types of data flow that a sub-program can have.
   - Understand the difference between passing a parameter by reference and by value.
   - Understand what happens to a local variable when the subprogram it's created in finishes

5. Does this resource fulfil a real need?

   In my opinion it does – understanding how parameters are passed in an out of subroutines are crucial to learning how to code properly.

6. Did you like learning from this resource?

   If I was in a big group, I could see this being very fun. The cards/code could have been a bit nicer to look at, it could even be customised/personalised a little more depending on the programming language that was being used by the school.

7. Did you learn from this resource? If yes, what?

   Instructions were very clear but this is clearly a group activity so it's difficult to understand fully what I was supposed to learn here, but it seems like it would be easy to understand how parameters are passed through by actually physically passing a 'parameter' to the next subroutine that needs it.

Activity Name: Data Modelling with ER Diagrams

1. What is the activity attempting to teach?

   How to identify entities, relationships and attributes within a system, by recreating a data model. It will also maybe help explain the difference between an entity and an instance and an attribute and its value.

2. How long did it take to complete?

   It claims it should take 1 period (~55 minutes)

3. What should students be able to do after successfully learning with the materials?
   - Given some example attributes be able to group them with particular entities
   - Given a set of values group them into specific instances of an entity.
   - Given a set of entities describe the relationships that may exist between them.
   - Define what the terms entity, attribute and relationship mean.

4. What are the learning objectives?
   - Understand what an entity and an attribute are
   - Understand how these are related to instances and values
   - Understand that different relationships can exist between different entities
   - Understand the purpose of an ER diagram
   - Appreciate that ER Diagrams are one way of creating an information model

5. Does this resource fulfil a real need?

   Yes – it is essential to learn about entity relationships in order to successfully model a system, it is important in order to create databases effectively and have an understanding of how the lower layers of a system works.

6. Did you like learning from this resource?

   Yes, it was very simple to use and understand, it helped convey the ideas of entities effectively by splitting up different objects into the right place. A little bit of interactivity that shows how ER diagrams and relationships work.

7. Did you learn from this resource? If yes, what?

   As I have already used ER diagrams before in the past, I had previous knowledge and didn't really need the resource in order to understand them, but by pretending to put all knowledge behind and ignoring what I know, I found the activities useful and a very good basic guide to be taken on further, the three exercises really gave a comprehensive understanding of the basics of ER diagrams.

Activity Name: Working with open data using files and records

1. What is the activity attempting to teach?

   Teach students about what open data is, different file formats, file handling, and database records

2. How long did it take to complete?

   Claims to take 3 single periods or two double periods.

3. What should students be able to do after successfully learning with the materials?
   - Understand and write programs to read in structured data from typical text file formats, e.g. CSV.
   - Design data structures using arrays and records to store structured data, e.g. an array of records
   - Know how to traverse a complex data structure in order to retrieve relevant information

4. What are the learning objectives?
   - Understand how a program can process complex structured data held in text files in order to derive specified information
   - Appreciate the similarities and differences of arrays of records within a programming language and a database table
   - Have an awareness of the phenomenon of open data and the value it has

5. Does this resource fulfil a real need?

   I don't suppose students really need to know about Open data. It is useful as a starter to reading in files and manipulating databases but due to it being written in Haggis it's a bit difficult to understand. Perhaps a basic pseudocode would have been easier to work with, or by using a programming language that a lot of people use would have bee more productive.

6. Did you like learning from this resource?

   It would be a great group exercise, it makes sense to have more than one person working on it, and I can imagine it works very well and would be enjoyable for groups. This is definitely for more advanced students, I can imagine it might even take longer than the allocated time recommended to complete.

7. Did you learn from this resource? If yes, what?

   I learned a bit more about open data, what Haggis is, how to manipulate databases in Haggis.

Activity Name: Server side scripting for web-based applications

1. What is the activity attempting to teach?

   This activity allows students to explore the client/server process used in web-based applications. They will simulate different interactions of data flowing between the browser and the webserver (specifically the HTTP server). They will start with simple examples and then gradually build up towards pages requiring a server side scripting engine and a database server. This can be used as either at the start of a block of learning as an overview or at the end to bring the different concepts together.

2. How long did it take to complete?

   2 single periods or 1 double period

3. What should students be able to do after successfully learning with the materials?

   Students should be able to explain how a web server responds to a request from a browser, and how the browser uses the HTML data. They should also be able to explain how a web server dynamically generates a page using a server-side script. They should also be able to explain how parameters can be passed with a URL to send data between the browser and the HTTP server. And finally explain how a database can be used to store information that appears on a web page.

4. What are the learning objectives?

   - Understand the relationship between the browser and the web server
   - Understand how files are requested from a web server, and how a server-side scripting language makes the contents of the pages dynamic
   - Understand the GET method of sending an item of data with a URL
   - Understand how a web server can use a server-side scripting language to interact with a database to retrieve information.

5. Does this resource fulfil a real need?

   Yes, learning about Networking and how web browsers interact with an internet server is very important for any students wishing to go on and further study anything in the topic of web development. It is good before they go to university to have an understanding of this.

6. Did you like learning from this resource?

   Yes, however I feel like students might forget or misunderstand what role they are playing in the activity due to just learning the concepts – which is why this activity may be better as a refresher for advanced higher students. Otherwise it gives a fun way of identifying how each part interacts with each other by having students re-enact parts of the system and by physically passing the images/webpages across the network.

7. Did you learn from this resource? If yes, what?

   Everything I already knew from before, but it’s a good introductory activity.

Activity Name: Box Variables (From STEM site)

1. What is the activity attempting to teach?

   An introduction to variables and how to assign variables, also sequencing and general programming.

2. How long did it take to complete?

   15-20 minutes, or a full period if using the extension exercises.

3. What should students be able to do after successfully learning with the materials?

   Be able to successfully use variables, understand how assignment works, successfully name variables, and generally understand how variables can be used in programming.

4. What are the learning objectives?
   - Understand variable creation
   - Understand variable naming techniques
   - Understand how variable assignment works
   - Understand the difference between values and variable names
   - Understand how variables can be assigned to other variables

5. Does this resource fulfil a real need?

   Yes, learning about variables is crucial for further learning in any programming language, and understanding how they work is important. Could be used to extend into explaining global and local variables, by maybe colour coordinating labels or such.

6. Did you like learning from this resource?

   Yes, it gave a very simple and easy understanding of how variables worked by giving a fun little group activity to work from. By involving other people and having physical representations of variables in the form of boxes, and values as sheets of paper, it allows the students to see exactly what is being passed and assigned to where. I imagine that if students were using this properly they would understand how variables worked perfectly.

7. Did you learn from this resource? If yes, what?

   Already knew all this information from school.

Activity Name: Spit-not-so (STEM)

1. What is the activity attempting to teach?

   Aims to give a deeper understanding of core computing topics such as computational thinking, human computer interaction and design, and data structures.

2. How long did it take to complete?

   20 – 30 minutes

3. What should students be able to do after successfully learning with the materials?

   Be able to start thinking conscientiously about graphical design when it comes to software, and not poorly design their user interfaces. Also start to consider appropriate data structures for their code. They will appreciate/understand the differences between the graphical user interface and a command line interface.

4. What are the learning objectives?
   - Understand general human computer interaction
   - Understand why graphical user interfaces are often easier to understand than command line interfaces
   - Introduce the idea of a data structure
   - Introduce how data structure choice makes a difference to how well a task can be done.
   - Also introduces the computational thinking ideas of translating problems and understanding people.

5. Does this resource fulfil a real need?

   I wouldn't say so, yes there is a need to understand how to design software appropriately, but a lot of that knowledge comes from common sense and from dealing with things outside of computing – like from games, toys, television and general graphical experience.

6. Did you like learning from this resource?

   By introducing a game and letting students play the game at different levels of design is very fun, it could frustrate the students at the start, and lets them see how design improvement would improve the playability of the game. It also helps you learn in a fun way of solving problems and coming up with algorithms that will help solve problems effectively.

7. Did you learn from this resource? If yes, what?

   How to play spit-not-so, and although I already knew about the importance of suitably designing a GUI, through use of the game it solidified my understanding of it.

General Results

Overall, of the activities I evaluated, a few key points can be made:

- Activities that involved heavy coding were generally more difficult to understand/teach and the concepts came across effectively, however I can imagine it might take a lot more unnecessary time for the teacher to understand how to teach it.
- Some activities involved games, but one was less focused than the others and didn't really help convey proper computing concepts.
- By allowing students to represent specific computing objects (like a student holding a box being a 'variable') helped hugely in understanding the concepts. By relating themselves to the concept itself, it helps convey the idea properly.
- Most of the activities took one hour or so to do, the more difficult coding activities taking a while longer.
- Even as an almost fourth year student studying computer science – I still actually sometimes learned some things while doing this! Proving not exactly everything is taught to you properly at school, and strengthens the need for such activities.
- All activities were meant to be for groups of people.

Using Elm to Introduce Algebraic Thinking to K-8 Students – Key points from text

After trial and error, we have found a Domain Specific Language for two-dimensional drawing embedded in the functional language Elm to be by far the most successful. Matching the semantics (and vocabulary) of drawing with stencils allows children to begin exploring with very little stumbling over new concepts.

In all cases, using a language in the ML family is important because

- pure functions match the child's inherent idea of tools, like stencils, pens and paint brushes
- it is easy to create a domain-specific language,
- there is minimal syntax to learn.

We prefer to call our approach "algebraic thinking", going back to the original definition of algebra as "taking apart and putting together". By focusing on shapes, children learn about recursive (tree) data structures by building increasingly complex pictures.

When we used Python, we needed to interleave girls and boys in the lab as a way to prevent friends from distracting each other from their programming task. However, after switching to Elm, we can encourage children to consult each other. Now, if we one team asks for help to make their shape blink, we will soon see half the class following suit in their animations.

The instructor shows students how to draw basic shapes on the centre of the collage. The next inquiry most students have is a logical progression from the current state: "How can we move the shape from the centre of the collage?" Like with filled, we use forward function application (|>) 1 to apply transformations to Shapes, thereby visually laying out the combinatorial nature of shape construction.

As a rule, tools meant for experts are not suitable for learners. This is certainly true of integrated development environments with lots of key-press-saving state. Many practitioners who see functional programming as the reserve of elite programmers have asked us why we use it for beginners. In this case, functional programming's fundamental advantages actually play out in the favour of beginners.

Once again, students suggested the obvious "copy and paste" approach, with flowerRed, flowerGreen, flowerBlue, etc. variables being created and then referenced in their collage. The volunteer instructor then emphasized the point that similar code should be reused, instead of duplicated. The ultimate motivation then becomes that learning how functions apply to Elm would benefit their more rapid and straightforward creation of new artwork and animations. Thus, the concept of functions was presented as a "tip" rather than a traditional lesson.

The discussions are presented as "help lines" where students can ask any questions about their code (working or broken), and mentors can send back answers or even links to differences between the student's code and code modified by the mentor. The mentor's modifications are to a private copy, not the student's code, or the code of another mentor.

This system is new, but students are already using it to ask questions after school hours, as well as to get answers faster during lab time when the on-site instructors are busy helping another student.

1. What is the activity attempting to teach?

   This activity aims to give K-8 students (ages 5-14, students who essentially go to a school that is a joined elementary and middle school) basic knowledge about variables, functions and general algebraic thinking (they prefer to think of it as going back to the original definition of algebra as "taking apart and putting together") by using a combination of a functional programming language Elm and a graphics package called GraphicsSVG.

2. How long did it take to complete?

   Never mentioned, but I imagine it was a workshop over a few days – a week

3. What should students be able to do after successfully learning with the materials?

   Students should be able to have a basic understanding of variables and functions, and how to reduce code from copying and pasting identical pieces of code to starting to recognise when to use functions effectively. Students should also be able to start thinking 'algebraically' by maybe taking problems and breaking them down into smaller pieces to be solves more effectively.

4. What are the learning objectives?

   - Understand what variables are and how they work
   - Understand when to use functions appropriately
   - Understand a basic knowledge of Elm and using a graphics package with Elm
   - Understand how to work together with other people to solve a problem

5. Does this resource fulfil a real need?

   Yes, this program teaches children the basic things they need to know about programming in an interactive way, and gets them thinking about computational concepts in a unique way. It is important for children to understand these concepts in order to continue and develop their skills in programming.

6. Did students like learning from this resource?

   From the report, it was clear that students enjoyed using this resource very much. By using a basic shape creator, it made the program very easy to understand, and by using a teacher demonstration first (to grab their attention) and then by allowing the students to look at the simplest problems (shape and colour) and then moving on to considering more difficult aspects of the program (functions, notifying and updating shapes) allows the students to work at a gradual pace in order to understand fully how things work rather than throwing them in at the deep end. It was said that for many students it was their first exposure to a programming language, and that seeing text produce a shape onscreen was very exciting and encouraged them to ask questions.

7. Did students learn from this resource? If yes, what?

   Students were able to map the “input-process-output” concept to three different ideas: an abstract concept, describing real-world processes, and then implementing the idea algebraically by programming. Many students made the connection to their maths classes where they had begun to talk about mathematical functions and lines but were motivated by being able to use the concept to improve their Elm graphics.

Hour Of Code – Minecraft Adventure

This was an online interactive Minecraft simulator, where users used Scratch like commands to move their Minecraft character about the screen and complete tasks like dig a block, shear a sheep and step over lava if necessary. The tutorial is designed for ages over six and up.

1. What is the activity attempting to teach?

   The introduction video outlines that the game will teach kids about commands, if cases and 'repeats' (loops) through the use of block building game Minecraft.

2. How long did it take to complete?

   Did take about one hour

3. What should students be able to do after successfully learning with the materials?

   The students should be able to understand why it's important to repeat steps that could become overused, such as 'walk forward turn left walk forward turn left walk forward turn left' could be put into a repeat loop of 3.

4. What are the learning objectives?
   - Understand how commands work in a simple environment
   - Understand how if/else commands work
   - Understand how loops can be used

5. Does this resource fulfil a real need?

   Yes, it is important for children to understand basic commands, ifs and loops in order to continue on with developing their skills as programmers.

6. Did you like learning from this resource?

   Yes, as an adult I have played the full version of Minecraft myself, so it was very interesting to see how Minecraft can be used in order to teach children about code. By using a game that millions of children around the world have played, it makes it relatable and allows children to manipulate characters that they have seen before in a different and fun way – all while learning a little bit about programming.

7. Did you learn anything? If yes, what?

   Everything I already knew, but I could see how it would be useful to teach children about this, it is very simple and easy to understand. The little video tutorials in between every couple of levels introduced the new topics very gently, and by seeing Jens (the chief developer for Minecraft) introduce a topic made it feel very official and correct.