Year 10 Apollo Archimedes Lesson plans
Programming Term 2

Contents
Lesson Plans .................................................................................................................. 2
Lesson 1 – Intro to Programming .................................................................................. 2
Lesson 2 – Design Challenge 1 ..................................................................................... 3
Lesson 3 – Introduction to the Programming Environment and programming ................. 4
Lesson 4 – Maths and Q basics ..................................................................................... 4
Lesson 5 – User Input ................................................................................................... 5
Lesson 6 – Arrays, Parallel arrays and 2D Arrays .......................................................... 5
Lesson 7 – Do Loops, For next ..................................................................................... 6
Lesson 8 – If then and select case statements ............................................................... 6
Lesson 9 – Design Challenge 2 (Chicken Cross puzzle) .................................................. 7
Lesson 10 – General programming .............................................................................. 7
Lesson 11 – String Manipulation .................................................................................. 8
Lesson 12 – Graphics 1 ................................................................................................ 8
Lesson 13 – Graphics 2 ................................................................................................ 9
Lesson 14 – Moving an object with a keyboard .............................................................. 9
Lesson 15 – Assignment Lesson 5 ............................................................................... 10
Lessons 16 – 17 Revision ............................................................................................ 10
Lesson 18 – Exam ....................................................................................................... 10
Lesson 19 – Design Challenge 3 (Flappy Bird) .............................................................. 11
Lesson 20 – Review of Term & Feedback from Exam .................................................... 11
Resources .................................................................................................................... 12
3.1 ideas for examples and students work ..................................................................... 12
4.1 Ideas for examples and students work ..................................................................... 12
5.1 Ideas for examples and students work ..................................................................... 12
6.1 Ideas for examples and students work ..................................................................... 12
7.1 Ideas for examples and students work ..................................................................... 12
8.1 Ideas for examples and students work ..................................................................... 12
9.1 Chicken Cross problems ......................................................................................... 12
  The Problem: ............................................................................................................... 12
  The Solution: ............................................................................................................... 13
11.1 Ideas for examples and students work ................................................................... 13
Exam ............................................................................................................................. 14
Text References ............................................................................................................. 19
Lesson Plans

Lesson 1 – Intro to Programming

<table>
<thead>
<tr>
<th>Topic</th>
<th>Time: 70 min</th>
<th>Lesson Number: 1</th>
<th>Author: Bishop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lesson Title:</td>
<td>Intro to Programming</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Lesson Intentions:
- Create a timeline
- Prioritising information & Summarising text
- Develop a basic knowledge of the history of aviation

Sequence/Strategies:
- **Warm up:** Get students to arrange themselves in age from youngest to oldest without speaking
- Spend a few minutes going through an outline of the term including dates for upcoming assessment (oral presentation so we will be building these skills throughout the term) and excursions before outlining today’s lesson.
- Hand out the demo timeline and text, discuss and make notes on the features, what has happened to the quantity of text and organisation. In what context would each format be useful?
- Arrange students in pairs or groups (maximum 3 in a group), each group will need a print out or access to the website (be careful they do not wander on the internet or they will find already prepared timelines) and paper to present their timelines on.
- Have them find the following language for their timeline. ALGAE, UNICODE, FORTRAN IV, BASIC, SNOBOL, TUTOR, PASCAL, SQL, COMMODORE BASIC, ICON, C++, GWBASIC, QUICKBASIC, LABVIEW, TURBO PASCAL OOP, VISUAL BASIC, BORLAND PASCAL, JAVA, PHP.
- Give a maximum of 20 minutes to read through the material and convert to their timeline.
- Allow students 5 min to wander around looking at the other groups timelines to see how much detail other groups have used.
- Ask student to decide which one or two will go on the wall as representative timelines for future reference.
- Introduce (or revise) Flowcharts and Pseudocode.
- Introduce QBASIC
- **Closure:** Students take a few minutes to complete the 3, 2, 1 go activity (3 things I have learnt, 2 things I enjoyed, 1 concern I have) and discuss with the person next to them.
- **Assign Homework,** if time go through using TED on the interactive whiteboard (they have seen this before so not necessary)

Resources/Materials/Weblinks:
- For teacher reference: [https://www.aiaa.org/HistoryTimeline/](https://www.aiaa.org/HistoryTimeline/)
- Resource **1.1 Timeline example** (one is supplied but a class relevant example would be more beneficial)
- About Programming : [http://www.bfoit.org/itp/Programming.html](http://www.bfoit.org/itp/Programming.html) (up to and including Programming using the English language )
- [http://outspeaking.com/words-of-technology/what-is-programming.html](http://outspeaking.com/words-of-technology/what-is-programming.html)
- PseudoCode [http://www.bfoit.org/itp/Pseudocode.html](http://www.bfoit.org/itp/Pseudocode.html) down to Looking back – ignore the Logo stuff
- Materials: Access to internet or pre-printed copies of information from the internet, poster/butchers paper, felt pens etc to create timeline, blue tack/tape to hang completed timelines.

Prior Knowledge:
- Students should already be comfortable with working in small groups and critiquing each other’s work, prior experience in note taking and summarizing text is also an advantage.

Homework/Assessment:
- Ask students to create a flowchart of a daily activity e.g getting out of bed and going to school. It could include loops like discovering it’s Saturday and returning to bed or Being sick and returning to bed.
Lesson 2 – Design Challenge 1

<table>
<thead>
<tr>
<th>Topic: Programming</th>
<th>Time: 70 min</th>
<th>Lesson Number: 2</th>
<th>Author: Bishop</th>
</tr>
</thead>
</table>

**Lesson Title:** Design Challenge 1: Humpty Dumpty Egg Drop

**Lesson Intentions:**
- Have students be given a set of parameters (Programming involves creating solutions from parameters)
- Be able to form their own groups
- Communicate orally within groups
- Analyse and discuss the problem
- Design a solution
- Attempt to explain how the prototype works

**Sequence/Strategies:**
- Adapt the egg drop challenges below to give the students a list of items that they need to build a device that will protect an egg from a second story fall.
- Have students work in groups to plan their approach; teacher should be careful not to help directly but to redirect students back to their plans if they go off track.
- Give clear time limits (30 mins should have them all done and tested)
- Introduce the Programming Process
  - Define the problem
  - Design the general logic of the program
  - Code the program
  - Test and Debug the Program
  - Document the Program

**Resources/Materials/Weblinks:**
- PMI charts: [https://learningplace.eq.edu.au/cx/resources/file/98953d22-08d0-c9c3-991d-9208f29754f5/1/index.html#pmi](https://learningplace.eq.edu.au/cx/resources/file/98953d22-08d0-c9c3-991d-9208f29754f5/1/index.html#pmi)
- PEEL paragraphs: [http://prezi.com/ebwgyb_2gyb/peel-paragraphs/](http://prezi.com/ebwgyb_2gyb/peel-paragraphs/)
- [http://www.cs.bham.ac.uk/~rxb/java/intro/2programming.html](http://www.cs.bham.ac.uk/~rxb/java/intro/2programming.html)

**Prior Knowledge:**
- Some experience in Predict, observe, explain (POE) activities, group work activities or analysis an advantage but not necessary

**Homework/Assessment:**
- A written analysis, beginning students can complete a ‘plus, minus, interesting’ (PMI) table, students with experience will need to write paragraphs explaining how or why their solution worked or failed and giving strengths and limitations of their design (they need to remember to PEEL in their paragraphs).
Lesson 3- Introduction to the Programming Environment and programming

<table>
<thead>
<tr>
<th>Topic: Programming</th>
<th>Time: 70 min</th>
<th>Lesson Number: 3</th>
<th>Author: Bishop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lesson Title: Intro to Programming Environment and programming</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Lesson Intentions:**
- Introduce the programming environment
- Tie in how flowcharting, pseudocode (planning) makes coding easier.

**Sequence/Strategies:**
- **Warm up** – In groups have students decide on their favourite game or app, a list of 10 or so. One of the favourites at this time was the flappy birds app – an easy to design scroller.
- Discuss how someone programmed the game using code similar to what will be learnt
- **Explicit teaching:** Demonstrate the QBASIC environment.
  - Introduce Variables (string and numbers) and DIM
  - Show constant and REM statements.
  - Print statement and print positioning
  - Demonstrate with short code sequences setting a variable, filling it and printing
- **Closure:** Discuss with the person beside what was covered today.

**Resources/Materials/Weblinks:**
- QBASIC Cheatsheet [http://www.eng.umd.edu/~nsw/chbe250/qbasic.htm](http://www.eng.umd.edu/~nsw/chbe250/qbasic.htm) (can get rid of the $ and % they aren’t necessary – its just a tie back to an older convention like the LET statement)
- [http://www.schoolfreeware.com/QBasic_Tutorial_1_-_Getting_Started_-_Free_Download_-_QB64.html](http://www.schoolfreeware.com/QBasic_Tutorial_1_-_Getting_Started_-_Free_Download_-_QB64.html)
- [http://www.schoolfreeware.com/QBasic_Tutorial_3_-_Print_Formatting_-_QB64.html](http://www.schoolfreeware.com/QBasic_Tutorial_3_-_Print_Formatting_-_QB64.html)
- [http://www.schoolfreeware.com/QBasic_Tutorial_4_-_Variables_And_Data_Types_-_QB64.html](http://www.schoolfreeware.com/QBasic_Tutorial_4_-_Variables_And_Data_Types_-_QB64.html)

**Prior Knowledge:**

**Homework/Assessment:**
- Review and look over the Qbasic cheat sheet to have a look at future lessons work.

Lesson 4 – Maths and Qbasics

<table>
<thead>
<tr>
<th>Topic: Programming</th>
<th>Time: 70 min</th>
<th>Lesson Number: 4</th>
<th>Author: Bishop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lesson Title: Maths and Qbasic</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Lesson Intentions:**
- To be able to use Maths functions in Qbasic
- Solve some Maths procedures using the computer

**Sequence/Strategies:**
- **Warm up:**
  - Before https://www.youtube.com/watch?v=Z-FKjql6NyQ - order of operations of maths song.

- **Explicit teaching:** Maths functions and using them, Division methods and MOD

- Do examples of the coding and have students create solutions to maths problems.
- Make sure students do problems by creating flowcharts and some psuedocode before programming
- **Closure:** Make a list of new key words from today’s lesson at the side of your page. Compare with the person next to you.

**Resources/Materials/Weblinks:**
- [http://www.schoolfreeware.com/QBasic_Tutorial_2_-_Printing_Math_-_QB64.html](http://www.schoolfreeware.com/QBasic_Tutorial_2_-_Printing_Math_-_QB64.html)
- [https://www.youtube.com/watch?v=rjhUVKhz-IQ&feature=youtu.be](https://www.youtube.com/watch?v=rjhUVKhz-IQ&feature=youtu.be)

**Prior Knowledge:**
- Basic knowledge of states of maths

**Homework/Assessment:**
- Continue working on glossary of motion
### Lesson 5 – User Input

<table>
<thead>
<tr>
<th>Topic: Programming</th>
<th>Time: 70 min</th>
<th>Lesson Number: 5</th>
<th>Author: Bishop</th>
</tr>
</thead>
</table>

**Lesson Title:** User Input  

**Lesson Intentions:**  
- Start work on assignment

**Sequence/Strategies:**  
- **Warm up:** ‘Count off’ students sit in a circle so they can see each other’s faces, tell them they are going to count as high as they can as a class, sounds easy doesn’t it? Anyone can say a number, but if two or more say the number at the same time we have to start over. No more than 3 seconds can go by between numbers *(Paterson, 2007).* Spend a maximum of 5 min on this.  
- **Explicit Teaching:** How to obtain a user input with INPUT, assign it to a variable and print out a result. Use this to add numbers and do other maths functions  
- **Closure:** Write a journal entry on their assignment progress today

**Resources/Materials/Weblinks:**  
- [http://www.schoolfreeware.com/QBasic_Tutorial_6_-_User_Input_-_QB64.html](http://www.schoolfreeware.com/QBasic_Tutorial_6_-_User_Input_-_QB64.html)

**Prior Knowledge:**  
- Earlier QBASIC work

**Homework/Assessment:** Have students flowchart and pseudocode the start screen to a video game where they have to enter their name and age and add it to variables.

### Lesson 6 – Arrays, Parallel arrays and 2D Arrays

<table>
<thead>
<tr>
<th>Topic: Programming</th>
<th>Time: 70 min</th>
<th>Lesson Number: 6</th>
<th>Author: Bishop</th>
</tr>
</thead>
</table>

**Lesson Title:** Arrays, Parallel arrays and 2D Arrays  

**Lesson Intentions:**  
- Develop an understanding of factors that create and influence drag

**Sequence/Strategies:**  
- **Warm up:** P.O.E. (describe activity to students, ask them to make a prediction, carry out activity while they observe then they try to explain what happened). Empty out a stapled teabag to make a cylinder and place on the table. Tell the students you are going to set fire to the top of the teabag. Get them to P.O.E. link to a website explaining this is in the weblinks below.  
- **Explicit teaching:** Arrays, Parallel Arrays and 2D Arrays.  
- **Closure:** Reflect over new programming syntax

**Resources/Materials/Weblinks:**  
- [http://www.schoolfreeware.com/QBasic_Tutorial_15_-_Arrays_-_QB64.html](http://www.schoolfreeware.com/QBasic_Tutorial_15_-_Arrays_-_QB64.html)  
- [http://www.schoolfreeware.com/QBasic_Tutorial_16_-_Parallel_Arrays_-_QB64.html](http://www.schoolfreeware.com/QBasic_Tutorial_16_-_Parallel_Arrays_-_QB64.html)  
- [http://www.schoolfreeware.com/QBasic_Tutorial_17_-_2_Dimensional_Array_-_Matrix_-_QB64.html](http://www.schoolfreeware.com/QBasic_Tutorial_17_-_2_Dimensional_Array_-_Matrix_-_QB64.html)

**Prior Knowledge:**  
- Previous experience at P.O.E. activities an advantage  
- Previous lessons in this unit a must

**Homework/Assessment:**  
- Review and go over the lessons of today
Lesson 7 – Do Loops, For next

<table>
<thead>
<tr>
<th>Topic: Programming</th>
<th>Time: 70 min</th>
<th>Lesson Number: 7</th>
<th>Author: Bishop</th>
</tr>
</thead>
</table>

**Lesson Title:** Do Loops

**Lesson Intentions:**
- Learn about making a program loop

**Sequence/Strategies:**
- Have students learn that a program often has to loop and stay active waiting for events or to complete a set amount of task.
- Do examples and have them complete looping tasks – stress the flowcharting to start them getting their logic straight.

**Resources/Materials/Weblinks:**
- [http://www.schoolfreeware.com/QBasic_Tutorial_10_Do_Loop__QB64.html](http://www.schoolfreeware.com/QBasic_Tutorial_10_Do_Loop__QB64.html)
- [http://www.schoolfreeware.com/QBasic_Tutorial_11_For_Loop__QB64.html](http://www.schoolfreeware.com/QBasic_Tutorial_11_For_Loop__QB64.html)

**Prior Knowledge:**
- Previous lessons a must

**Homework/Assessment:**
- Review of the days work

---

Lesson 8 – If then and select case statements

<table>
<thead>
<tr>
<th>Topic: Programming</th>
<th>Time: 70 min</th>
<th>Lesson Number: 8</th>
<th>Author: Bishop</th>
</tr>
</thead>
</table>

**Lesson Title:** If then else and select case statements

**Lesson Intentions:**
- Move on with programming and let the program detect a response and act on it depending on set parameters

**Sequence/Strategies:**
- **Warm up:** White board quiz (give each student a whiteboard, pen and eraser, as a question give approximately 10 sec to write down their answer and then they all hold up their boards together) Good to gauge how they are learning in the topic. List of whiteboard questions in Resource 8.1
- **Explicit teaching:** How if then else works and how it ties to flowcharting.
- Do examples of problems with flowcharts first to show how the logic is worked and how a program can choose an outcome in response.
- Do examples of Select case
- **Closure:** Review the lesson and see how the new syntax has added to our library of understanding

**Resources/Materials/Weblinks:**
- Mini white boards, pens and erasers.
- Notes in Resource 8.1 and in resource 8.2

**Prior Knowledge:**
- Use of white boards in the past an advantage

**Homework/Assessment:**
- Assignment
### Lesson 9 – Design Challenge 2 (Chicken Cross puzzle)

<table>
<thead>
<tr>
<th>Topic: Programming</th>
<th>Time: 70 min</th>
<th>Lesson Number: 9</th>
<th>Author: Bishop</th>
</tr>
</thead>
</table>

**Lesson Title:** Design Challenge 2 (Chicken cross Problem)

**Lesson Intentions:**
- Follow a problem and come up with a solution
- Work in groups and communicating orally within groups
- Represent the solution in flowchart

**Sequence/Strategies:**
- Assign pairs (try to do this randomly so students are not always working with the same person)
- Look at the Chicken cross problem in 9.1
- Have students work out the solution then write down the steps
- Have students synthesis this into a flowchart that could be used to program the steps
- Create psuedocode
- Students could program this if they have time
- Finish with a written analysis, beginning students can complete a ‘plus, minus, interesting’ (PMI)table, students with experience will need to write paragraphs reflecting on how well their solution and flowchart meet the challenge question clearly stating strengths and limitations (they need to remember to PEEL in their paragraphs).

**Resources/Materials/Weblinks:**
- Paper, pens, pencils, rulers etc
- Stopwatch or timekeeper

**Prior Knowledge:**
- flowcharting

**Homework/Assessment:**
- review psuedocode

### Lesson 10 – General programming

<table>
<thead>
<tr>
<th>Topic: Programming</th>
<th>Time: 70 min</th>
<th>Lesson Number: 10</th>
<th>Author: Bishop</th>
</tr>
</thead>
</table>

**Lesson Title:** General programming

**Lesson Intentions:**
- Consolidate the last few lessons and let students synthesis the information into usable knowledge.

**Sequence/Strategies:**
- **Warm up:** Start with a quick discussion on how their programming is going. Are they finding any aspects difficult?
- **Focus** on some problems to program getting used to the advances we have made.
- **Closure:** Write a journal entry on their assignment progress today

**Resources/Materials/Weblinks:**
- 

**Prior Knowledge:**
- Previous lessons and flowcharting

**Homework/Assessment:**
- 
# Lesson 11 – String Manipulation

<table>
<thead>
<tr>
<th>Topic: Programming</th>
<th>Time: 70 min</th>
<th>Lesson Number: 11</th>
<th>Author: Bishop</th>
</tr>
</thead>
</table>

**Lesson Title:** String Manipulation

**Lesson Intentions:**
- Be able to use a program to alter a string, find parts in a string etc.

**Sequence/Strategies:**
- Have students learn to manipulate a string.
- Use examples to demonstrate the principles.
- Have students work through a variety of problems to concrete understanding

**Resources/Materials/Weblinks:**
- [http://www.schoolfreeware.com/QBasic_Tutorial_22_-_String_Manipulation_-_QB64.html](http://www.schoolfreeware.com/QBasic_Tutorial_22_-_String_Manipulation_-_QB64.html)
- [http://www.schoolfreeware.com/QBasic_Tutorial_24_-_ASCII_Program_-_Separating_Letters_Numbers_And_Other_Characters_-_QB64.html](http://www.schoolfreeware.com/QBasic_Tutorial_24_-_ASCII_Program_-_Separating_Letters_Numbers_And_Other_Characters_-_QB64.html)
- [http://www.schoolfreeware.com/QBasic_Tutorial_23_-_Palindrome_Program_-_QB64.html](http://www.schoolfreeware.com/QBasic_Tutorial_23_-_Palindrome_Program_-_QB64.html)

**Prior Knowledge:**
- Previous lessons should all be utilised in this activity.

**Homework/Assessment:**
- Review the syntax of today

---

# Lesson 12 – Graphics 1

<table>
<thead>
<tr>
<th>Topic: Programming</th>
<th>Time: 70 min</th>
<th>Lesson Number: 12</th>
<th>Author: Bishop</th>
</tr>
</thead>
</table>

**Lesson Title:** Screen manipulation

**Lesson Intentions:**
- Understand how to create computer graphics.

**Sequence/Strategies:**
- **Warm up:** ‘Count off’ students sit in a circle so they can see each other’s faces, tell them they are going to count as high as they can as a class, sounds easy doesn’t it? Anyone can say a number, but if two or more say the number at the same time we have to start over. No more than 3 seconds can go by between numbers (Paterson, 2007). Spend a maximum of 5 min on this.
- **Explicit Teaching:** Run through the basics of computer graphics – creating shapes and colouring them
- **Closure:** Make a list of new things learnt from today’s lesson at the side of your page. Compare with the person next to you.

**Resources/Materials/Weblinks:**
- [http://www.schoolfreeware.com/QBasic_Tutorial_28_-_Drawing_Part_1_-_QB64.html](http://www.schoolfreeware.com/QBasic_Tutorial_28_-_Drawing_Part_1_-_QB64.html)
- [http://www.schoolfreeware.com/QBasic_Tutorial_29_-_Drawing_Part_2_-_QB64.html](http://www.schoolfreeware.com/QBasic_Tutorial_29_-_Drawing_Part_2_-_QB64.html)
- Selection of key words from previous lessons

**Prior Knowledge:**
- Previous lessons from this unit needed

**Homework/Assessment:**
- Assignment
**Lesson 13 – Graphics 2**

<table>
<thead>
<tr>
<th>Topic: Programming</th>
<th>Time: 70 min</th>
<th>Lesson Number: 13</th>
<th>Author: Bishop</th>
</tr>
</thead>
</table>

**Lesson Title:** Graphics 2

**Lesson Intentions:**
- Analyse oral presentations to make a list of key features

**Sequence/Strategies:**
- **Warm up:** ‘Time Machine’ ([Paterson, 2007](http://www.schoolfreeware.com/QBasic_Tutorial_30_-_Drawing_Part_3_-_QB64.html)) Ask students to close their eyes and imagine they are entering a time machine. What time are they going to travel to, forwards or backwards in time? Who are they with? What do they see? Guide them through this for a maximum of three minutes. Ask the students to share what they imagined with the person next to them. Ask some students to stand up and share with the class (impromptu oral presentation)
- Further programming of the graphics interface. Use examples and have students use code.
- **Closure:** Reflect on today’s lesson.

**Resources/Materials/Weblinks:**
- [http://www.schoolfreeware.com/QBasic_Tutorial_30_-_Drawing_Part_3_-_QB64.html](http://www.schoolfreeware.com/QBasic_Tutorial_30_-_Drawing_Part_3_-_QB64.html)

**Prior Knowledge:**
- Having completed oral presentation in the past is beneficial but not mandatory

**Homework/Assessment:**
- Assignment/Finish Journal

---

**Lesson 14 – Moving an object with a keyboard**

<table>
<thead>
<tr>
<th>Topic: Programming</th>
<th>Time: 70 min</th>
<th>Lesson Number: 14</th>
<th>Author: Bishop</th>
</tr>
</thead>
</table>

**Lesson Title:** Moving an object with the keyboard

**Lesson Intentions:**
- Explain what turbulence is and how it affects flight
- Explain what a stall is and how it affects flight

**Sequence/Strategies:**
- Explicit Teaching
- Students
- **Closure:** Make a list of new key words from today’s lesson at the side of your page. Compare with the person next to you.

**Resources/Materials/Weblinks:**
- Book computer lab or laptops

**Prior Knowledge:**
- Previous experience at creating questions for others an advantage.
- Previous lessons from this unit needed

**Homework/Assessment:**
- Assignment
## Lesson 15 – Assignment Lesson 5

<table>
<thead>
<tr>
<th>Topic: Programming</th>
<th>Time: 70 min</th>
<th>Lesson Number: 15</th>
<th>Author: Bishop</th>
</tr>
</thead>
</table>

**Lesson Title:** Programming  

**Lesson Intentions:**  
- Revise and consolidate graphics lessons until now  

**Sequence/Strategies:**  
- **Warm up:** ‘Count off’ students sit in a circle so they can see each other’s faces, tell them they are going to count as high as they can as a class, sounds easy doesn’t it? Anyone can say a number, but if two or more say the number at the same time we have to start over. No more than 3 seconds can go by between numbers (Paterson, 2007). Spend a maximum of 5 min on this.  
- Work on examples to make sure they can use all the variety of code up till now  
- **Closure:** Write a journal entry on their progress today

**Resources/Materials/Weblinks:**  

**Prior Knowledge:**  

**Homework/Assessment:**  
- Assignment

## Lessons 16 – 17 Revision

Note this may take more lessons depending on class size

<table>
<thead>
<tr>
<th>Topic: Programming</th>
<th>Time: 70 min each</th>
<th>Lesson Number: 16-18</th>
<th>Author: Bishop</th>
</tr>
</thead>
</table>

**Lesson Title:** Assignment Hand in  

**Lesson Intentions:**  
- Assessment  
- Critiquing and giving feedback to others  

**Sequence/Strategies:**  
- All students must submit hard copies of their presentation along with their annotated bibliography and Journal on entry to the room.  
- Use a random selection process to determine the order students present  
- It may be useful to already have PPT’s or videos that students are using on a presentation laptop.  
- Students give feedback to each other using the template supplied (Resource 16.1). You may like to do this in pairs. They must write their name on the feedback sheet and own their comments.

**Resources/Materials/Weblinks:**  
- Book computer lab or laptops

**Prior Knowledge:**  
- Previous experience at self or peer evaluation an advantage

**Homework/Assessment:**  
- Nil

## Lesson 18 – Exam

<table>
<thead>
<tr>
<th>Topic: Programming</th>
<th>Time: 70 min</th>
<th>Lesson Number: 18</th>
<th>Author: Bishop</th>
</tr>
</thead>
</table>

**Lesson Title:** Exam lesson  

**Lesson Intentions:**  
- Test the unit  

**Sequence/Strategies:**  

**Resources/Materials/Weblinks:**  
- Exam in resource 18.1

**Prior Knowledge:**  
- Previous unit

**Homework/Assessment:**  
- Nil
## Lesson 19 – Design Challenge 3 (Flappy Bird)

<table>
<thead>
<tr>
<th>Topic: Programming</th>
<th>Time: 70 min</th>
<th>Lesson Number: 19</th>
<th>Author: Bishop</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lesson Title:</strong> Assignment Lesson 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Lesson Intentions:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Work in groups and communicating orally within groups</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Make predictions/hypothesis</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Design the program</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Analyse the outcomes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Sequence/Strategies:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Have students use their skills to date to create the flappy bird program – non scrolling just bird from one edge to the other.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• They won’t be able to do sprite edge detection but can still have a go at designing the layout and the physics</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• They should be able to use some conditions to check where the position of the object is and where the pole graphics are to give a gameover.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Compete to see who has the better design.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Resources/Materials/Weblinks:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Prior Knowledge:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Prior programming</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Homework/Assessment:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Finish with a written analysis, beginning students can complete a ‘plus, minus, interesting’ (PMI)table, students with experience will need to write paragraphs reflecting on how well their program design meet the challenge goal clearly stating strengths and limitations (they need to remember to PEEL in their paragraphs).</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Lesson 20 – Review of Term & Feedback from Exam

<table>
<thead>
<tr>
<th>Topic: Programming</th>
<th>Time: 70 min</th>
<th>Lesson Number: 20</th>
<th>Author: Bishop</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lesson Title:</strong> Review of term &amp; Feedback from assessment</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Lesson Intentions:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Create a glossary of terms</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Reflect on learning from this term</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Sequence/Strategies:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• <strong>Warm up:</strong> ‘Count off’ students sit in a circle so they can see each other’s faces, tell them they are going to count as high as they can as a class, sounds easy doesn’t it? Anyone can say a number, but if two or more say the number at the same time we have to start over. No more than 3 seconds can go by between numbers (<a href="http://example.com">Paterson, 2007</a>). Spend a maximum of 5 min on this.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Students need to make a list of all the programming words they collected over the term</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Resources/Materials/Weblinks:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Book computer lab or laptops if you would like students to create a digital glossary. They then need to turn this into a glossary by defining the terms in their own words. They can use their notes and internet research if available. Ask students to check each other’s.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Spend time going through feedback from their exams. You may like to speak to them one on one while the class works on the glossary.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Finish of the lesson with a reflection on the term in their journal. They should make a point of setting some goals and outlining strategies in this journal entry.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Prior Knowledge:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Use and creation of a glossary</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Journal writing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Homework/Assessment:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Nil</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Resources

3.1 ideas for examples and students work
1. set their own name and print it to the screen.
2. Say hello
3. Say hello world
4. Print hello and have the student add something from the variable

4.1 Ideas for examples and students work
1. Adding or subtracting numbers and print the output
2. Multiply and divide numbers and print the output
3. Find the square root of a number and print the output
4. Square a number and print the output
5. Find the sin and cos of a number (remember it uses radians)

5.1 Ideas for examples and students work
1. Have the student enter something and do any of the examples above.

6.1 Ideas for examples and students work
1. Enter different numbers into an array and print the numbers out
2. Enter the names of 3 friends and have the program say hi to them all
3. Enter numbers and find the sin and cos of them (may have to teach the conversion from degrees)
4. 

7.1 Ideas for examples and students work
1. Do any of the previous but have the do loop or for next loop go back to get the next input

8.1 Ideas for examples and students work
1. create a game where the student enters a number to guess the number a computer has chosen (introduce the random function)
2. Ask for first, last and middle name then assemble them.
3. Ask for 3 favourite things then print “Fred like”.....

9.1 Chicken Cross problems

The Problem:

A man has to get a fox, a chicken, and a sack of corn across a river.

He has a rowboat, and it can only carry him and one other thing.

If the fox and the chicken are left together, the fox will eat the chicken.
If the chicken and the corn are left together, the chicken will eat the corn.

How does the man do it?

The Solution:

The man and the chicken cross the river, (the fox and corn are safe together), he leaves the chicken on the other side and goes back across.

The man then takes the fox across the river, and since he can't leave the fox and chicken together, he brings the chicken back.

Again, since the chicken and corn can't be left together, he leaves the chicken and he takes the corn across and leaves it with the fox.

He then returns to pick up the chicken and heads across the river one last time.

11.1 Ideas for examples and students work

1. Find letters in a string
2. Count the s’s in Mississippi
3. Break a work out of a string like hi from hellohigoodbye
Exam

1. Programming Process (D Standard)
   a) List the steps for in the programming process
   b) Describe what pseudocode is
   c) Describe the Debugging process.

2. What are the flow chart symbols for (D Standard)
   a) A decision
   b) Program start
   c) An input

3. Write the proper syntax for the following: (C Standard)
   a) Do Until
   b) For
   c) Case

4. How many variable spaces are created by (C Standard)
   a) Dim a as integer
   b) Dim a(3) as integer
   c) Dim a(3,3) as integer

5. Write a short code to get hello from the following statement and print it (C standard)
   ourword="whathelloyesno"

6. Complete the Flowchart (B Standard)

![Flowchart]

7. What has the programmer done wrong? (B Standard)
   ‘print a letter supplied by the user the ‘amount of time the user wishes
DIM a AS INTEGER
DIM b AS STRING
DIM c
CLR
DO
INPUT, Enter a letter ,a
INPUT, Enter a number,b
FOR c = 1 TO b
    PRINT a
END IF
NEXT

8. Write a flowchart and code for the following problems.
   a) The computer randomly chooses a number. A user has 3 goes to input the correct number, the computer tells the user if it’s higher, lower or they guessed the number (B Standard)

   b) When a user input a number, print the sin, cos and tan of the number and the number before and the number below, The output is in the form of a table with headings (A Standard) ‘they need to remember that qbasic uses radians

   c) The conversion from Celsius to Fahrenheit is 1.8C + 32. Print a table with a label at the top, showing temperatures from 20° C to -20°C celsius with the matching Fahrenheit temperature showing both temperatures with headings using . (A Standard)
<table>
<thead>
<tr>
<th>Dimension</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge and application</td>
<td>The student work has the following characteristics: accurate and consistent recall and comprehensive explanation of engineering knowledge, mathematical concepts and techniques relevant to complex engineering situations, across the areas of study. Critical elements of engineering problems are clearly identified and prioritised. Discerning selection and correct and efficient application of engineering knowledge, mathematical concepts and techniques to complex familiar and unfamiliar engineering problems.</td>
<td>The student work has the following characteristics: accurate recall and explanation of engineering knowledge, mathematical concepts and techniques relevant to engineering situations, across the areas of study. Significant elements of engineering problems are identified. Appropriate selection and correct application of engineering knowledge, mathematical concepts and techniques to complex familiar or simple unfamiliar engineering problems.</td>
<td>The student work has the following characteristics: recall and explanation of engineering knowledge, mathematical concepts and techniques relevant to engineering situations. Obvious elements of engineering problems are identified. Selection and application of engineering knowledge, mathematical concepts and techniques to simple familiar engineering problems.</td>
<td>The student work has the following characteristics: recall of engineering knowledge related to engineering situations. Basic aspects of engineering problems are recognised. Application of basic engineering knowledge to situations.</td>
<td>The student work has the following characteristics: recall of basic engineering facts. Use of basic engineering knowledge.</td>
</tr>
<tr>
<td>Dimension</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------</td>
</tr>
<tr>
<td>Investigative and analytical</td>
<td>The student work has the following characteristics: effective</td>
<td>The student work has the following characteristics: correct interpretation and</td>
<td>The student work has the following characteristics: interpretation and analysis of engineering data 8a solutions based</td>
<td>The student work has the following characteristics: solutions related to engineering principles or techniques are proposed</td>
<td>The student work has the following characteristics: factual statements are made about data ideas related to aspects of the problems are suggested</td>
</tr>
<tr>
<td>processes</td>
<td>interpretation and thorough analysis of relevant engineering data</td>
<td>detailed analysis of obvious relevant engineering data</td>
<td>and explanation of basic engineering data</td>
<td>and engineering principles or techniques are proposed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>efficient and mathematically validated engineering solutions</td>
<td>effective solutions based on engineering principles and techniques, and including</td>
<td>solutions based on engineering principles and techniques are</td>
<td>solutions related to engineering principles or techniques are</td>
<td></td>
</tr>
<tr>
<td></td>
<td>based on engineering principles and techniques are proposed</td>
<td>mathematical calculations, are proposed</td>
<td>proposed</td>
<td>proposed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>solutions are analysed in depth and detail from multiple</td>
<td>solutions are analysed in detail to identify relevant engineering principles</td>
<td>solutions are analysed in relation to engineering principles</td>
<td>solutions are analysed in relation to engineering principles</td>
<td></td>
</tr>
<tr>
<td></td>
<td>perspectives to identify relevant engineering principles</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>optimal prototypes and/or models that validate solutions are</td>
<td>effective prototypes and/or models that test solutions are developed and modified.</td>
<td>workable prototypes and/or models that test solutions are</td>
<td>prototypes or models that test aspects of solutions are produced</td>
<td></td>
</tr>
<tr>
<td></td>
<td>developed and refined.</td>
<td></td>
<td>developed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8b 8c</td>
<td></td>
<td>7</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Dimension</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------------------------------------------------------------------</td>
<td>------------------------------------------------------------------</td>
<td>------------------------------------------------------------------</td>
<td>------------------------------------------------------------------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td>Evaluation and technical communication</td>
<td>The student work has the following characteristics: comprehensive evaluation of solutions in relation to the critical elements of engineering problems</td>
<td>The student work has the following characteristics: considered evaluation of solutions in relation to the significant elements of engineering problems 7 8a</td>
<td>The student work has the following characteristics: evaluation of solutions in relation to obvious elements of engineering problems 4</td>
<td>conclusions and recommendations are based on investigations</td>
<td>The student work has the following characteristics: comparison of solutions in relation to engineering problems</td>
</tr>
<tr>
<td></td>
<td>valid, well-reasoned conclusions and recommendations based on investigations and justified by relevant engineering knowledge and data</td>
<td>valid conclusions and recommendations based on investigations and supported by engineering knowledge or data 6</td>
<td>conclusions and recommendations are based on investigations</td>
<td>conclusions are stated and recommendations made</td>
<td></td>
</tr>
<tr>
<td></td>
<td>effective organisation and succinct presentation of information in the most appropriate modes relevant to engineering situations</td>
<td>logical organisation and clear presentation of information in appropriate modes relevant to engineering situations 8a</td>
<td>organisation and presentation of information in modes relevant to engineering situations 3 5</td>
<td>presentation of engineering information</td>
<td>presentation of some information related to engineering.</td>
</tr>
<tr>
<td></td>
<td>consistent and proficient demonstration of technical and spatial literacy through the discerning selection and appropriate use of a wide range of literacy forms. 8b 8c</td>
<td>demonstration of technical and spatial literacy through the selection and use of a range of literacy forms. 7</td>
<td>demonstration of technical and spatial literacy. 5</td>
<td>demonstration of technical or spatial literacy.</td>
<td></td>
</tr>
</tbody>
</table>
Text References