NORTHCOTE HIGH SCHOOL

NAME: _____

TEACHER: **Mr. Koopmans**

COMPUTING UNIT 2
OUTCOME 1 (SAC 4)

Due: Monday 13 August 2018
(Start of class)

PORTFOLIO QUESTIONS

Structure of book

Folio	*Number of questions*	*Number of questions to be answered*	*Number of marks*
1	6	6	20
2	6	6	20
3	6	6	20
4	6	6	20
5	6	6	20
		Total	100

Outcome 1:
Design working modules in response to solution requirements, and use a programming or scripting language to develop the modules.

Task:
Portfolio of programming module solutions
This task will be marked out of 100. It will contribute 100% of the marks allocated for this outcome.

Folio 0 (Example) – Fizz Buzz **(No marks – example)**

To help students learn about division, Northcote Primary School plays a game called Fizz Buzz. In the game, students take turns counting from 1 to 50, replacing any number divisible by 3 with “Fizz”, numbers divisible by 5 with “Buzz”, and numbers divisible by both with “Fizz Buzz”.

The school would like to extend their game to work with any two numbers, and to count to any maximum number. Create a module that will help teachers check the correct responses by allowing them to enter the “Fizz” and “Buzz” numbers, and the maximum number.

Required

1. Use a design tool to show how the user interface will appear. Annotate the diagram to show formats and conventions used.

4 marks

2. Use a design tool to show the processing steps that will occur in the module.

4 marks

3. Use a design tool to show all of the variables used within the module.

2 marks

4. Design a series of tests, including the development of test data that will test if each solution requirement is working as expected.

3 marks

5. Develop the solution using an appropriate programming or scripting language.

6 marks

6. Annotate the test table with actual results. Indicate how many attempts were made before the expected results were actually achieved (if ever).

1 mark

Total: 20 marks

Task

Folio 1 - Barry's Fishmonger **20 marks**

Big Bad Barry sells fresh fish at the local market. Barry sells fish at the follow prices:

Fish type	Price per Kg
Barramundi Wild	$9.80
Flathead Tails	$10.99
Rockling	$9.70
Snapper Fillets	$9.80
Gummy Shark (flake)	$8.50

In any one purchase a customer can purchase any one type of fish, up to 20Kg. Businesses are the bulk of Barry's purchasers and in order to keep them, they receive a 10% discount off the total purchase price.

For example, Mary, a non-business customer would like 1.5Kg of Rockling fillets for a family gathering. 1.5 x $9.70 = $14.55

Create a module that will enable customers to input the type and weight of the fish they want, if they are a business customer or not and calculate the total price of their order.

Required

1. Use a design tool to show how the user interface will appear. Annotate the diagram to show formats and conventions used.

4 marks

2. Use a design tool to show the processing steps that will occur in the module.

4 marks

3. Use a design tool to show all of the variables used within the module.

2 marks

4. Design a series of tests, including the development of test data that will test if each solution requirement is working as expected.

3 marks

5. Develop the solution using an appropriate programming or scripting language.

6 marks

6. Annotate the test table with actual results. Indicate how many attempts were made before the expected results were actually achieved (if ever).

1 mark

Total: 20 marks

Task

Folio 2 – Item Generator **20 marks**

You have been asked by a friend to create a program to allow them to create an item list of weapons for each player in their weekly Dungeons and Dragons game. Each player is allowed to have a maximum of 5 *unique* items. The list of items is below.

Functional requirements

- Imitate a dice roll by selecting a random number between 1 and 25.
- Select an item associated with the number rolled.
- Select up to 5 unique items.
- Reset the list at any time.
- Display the value of a dice roll and the corresponding item to be added to the list.
- Display only the items selected by that player.
- Each item selected by the player can only be selected once.

Item List

Broadsword, Shield, Helmet, Rope, Battle Axe, Daggar, Net, Sica, Scythe, Spear, Mask, Chest Plate, Boots, Backpack, Chain, Mace, Gloves, Chain mail, Bow & Arrows, Crossbow, Bear Claw, Scimitar, Longsword, Samurai Sword, Pike

Required

1. Use a design tool to show how the user interface will appear. Annotate the diagram to show formats and conventions used.

4 marks

2. Use a design tool to show the processing steps that will occur in the module.

4 marks

3. Use a design tool to show all of the variables used within the module.

2 marks

4. Design a series of tests, including the development of test data that will test if each solution requirement is working as expected.

3 marks

5. Develop the solution using an appropriate programming or scripting language.

6 marks

6. Annotate the test table with actual results. Indicate how many attempts were made before the expected results were actually achieved (if ever).

1 mark

Total: 20 marks

Task

Folio 3 – Times Table Creator **20 marks**

The Grade 3 teacher at Computerland Primary School has asked you to come up with a way to help students learn the first ten calculations of the times tables for any number the student wants to learn. They would like a system developed where the student types in a number, chooses how many calculations they would like to see (up to 10) and then display the output as a sentence. For example, if the student wants to learn the first three calculations of the six times table, your system should display.

1 times 6 is 6
2 times 6 is 12
3 times 6 is 18

Required

1. Use a design tool to show how the user interface will appear. Annotate the diagram to show formats and conventions used.

4 marks

2. Use a design tool to show the processing steps that will occur in the module.

4 marks

3. Use a design tool to show all of the variables used within the module.

2 marks

4. Design a series of tests, including the development of test data that will test if each solution requirement is working as expected.

3 marks

5. Develop the solution using an appropriate programming or scripting language.

6 marks

6. Annotate the test table with actual results. Indicate how many attempts were made before the expected results were actually achieved (if ever).

1 mark

Total: 20 marks

Task

Folio 4 - Number Guesser **20 marks**

"Marvin the Magnificent*" has been developing a new (but not overly exciting) magic trick. He believes that he can guess any number between 0 and 999 in ten guesses or less by only being told if his guess is high or low.
In order to perfect this level of magical skill, he has requested that you write him a program that will choose a random number less than 1000. He then types his guess into a box and a message appears to say if his number is too high or too low. If he guesses the right answer within 10 guesses, a message should appear saying he has won. If he hasn't guessed the correct answer within 10 guesses, then a message should say that he has lost. In order to maintain the incredible suspense, the system should also display the number of guesses he has used as well as the number of guesses remaining, updated after each guess.

**"Marvin the Magnificent" is not a real magician and bears no similarities to magicians alive or dead, whether or not they are actually funny, magical or downright scary.*
No numbers were injured in the creation of this task.

Required

1. Use a design tool to show how the user interface will appear. Annotate the diagram to show formats and conventions used.

4 marks

2. Use a design tool to show the processing steps that will occur in the module.

4 marks

3. Use a design tool to show all of the variables used within the module.

2 marks

4. Design a series of tests, including the development of test data that will test if each solution requirement is working as expected.

3 marks

5. Develop the solution using an appropriate programming or scripting language.

6 marks

6. Annotate the test table with actual results. Indicate how many attempts were made before the expected results were actually achieved (if ever).

1 mark

Total: 20 marks

Task

Folio 5 – Fibonacci and friends **20 marks**

Sammy Sequence loves number sequences, in particular the Fibonacci sequence (e.g. 1,2,2,3,5,8…..) . Sammy loves the Fibonacci sequence so much, that Sammy has asked for your help to create a program that will take any 2 numbers and create a Fibonacci sequence for any number of iterations entered into the system.

Required

1. Use a design tool to show how the user interface will appear. Annotate the diagram to show formats and conventions used.

4 marks

2. Use a design tool to show the processing steps that will occur in the module.

4 marks

3. Use a design tool to show all of the variables used within the module.

2 marks

4. Design a series of tests, including the development of test data that will test if each solution requirement is working as expected.

3 marks

5. Develop the solution using an appropriate programming or scripting language.

6 marks

6. Annotate the test table with actual results. Indicate how many attempts were made before the expected results were actually achieved (if ever).

1 mark

Total: 20 marks

Marking Rubric

VCE Computing UNIT 2, Outcome 1: Programming portfolio

Student Name: _____ Folio Number:

Task/Marks	0	1	2	3	4	5	6
User Interface or Object Descriptors (4 marks)	No design of user interface	Some design shown	Design mostly present	Design mostly complete	Design fully complete		
Processing steps (4 marks)	No design showing processing steps	Some steps shown	Most steps shown	All steps shown but contains a logic error	All steps shown are correct		
Variables (2 marks)	No design showing variables	Some variables identified	All variables identified				
Programming Module (6 marks)	No code shown	An attempt at coding has been made	Some of the lines of code required have been completed	Most lines of code required are shown	Code completed but contains 2 or more errors	Code completed but contains one error	Code fully completed and no errors present
Testing (4 marks)	No testing shown	Some tests created	Most	All	Tests and test data created for each solution requirement		
Sub-Total:							
Total (out of 20)							

Comments:

Teacher: **Date:**