Year 7 Term 2 Assessment Notification

Canberra Grammar School

Subject:	**Digital Technologies**
Task:	**Python Programming Challenge**
Due Dates:	**8:45am Monday Week 3 – Grok Learning Modules 1-3** **8:45am Monday Week 5 – Grok Learning Modules 4-6** **8:45am Monday Week 6 – Python Programming Challenge in ED STEM**
Weight:	**40%**

Content of the Task

During Term 2, you will be using Grok Learning to work through the first 6 modules of the course: **Introduction to Programming (Python).**

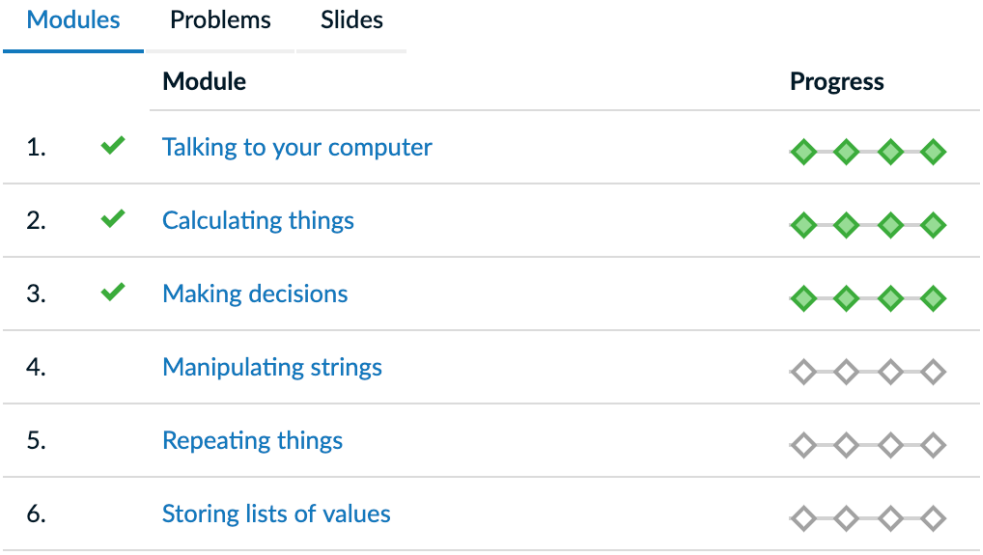
You are also required to complete a **Python Programming Challenge in ED STEM**, which is based on the programming concepts covered in Grok Learning.

Requirements of the Task

Grok Learning

For this part, you simply need to progress through the course before the due dates.

You can track your progress here: https://groklearning.com/course/intro-python-1/



Python Programming Challenge

For this part, you will need to access the Python Programming Challenges in ED STEM, by clicking on the 'book' icon (shown below), and follow the instructions there. You will need to complete the code for 6 python lessons, to demonstrate your understanding of each module shown above.

Submission instructions

Grok Learning

You simply need to progress through each module and ensure that you have successfully completed and marked the Grok Learning activities, before the due date. (1 mark awarded for each green diamond)

Python Programming Challenges

Similar to Grok Learning, you are able to run your python code and mark your submission, to see if you produce the correct output. There are some key differences, however:

- You will have 6 lessons to complete (one based on each Grok Learning Module)
- You will need to use the terminal window in ED STEM to run and test your python scripts.
- **THERE WILL BE NO MARKS DEDUCTED FOR INCORRECT SUBMISSIONS**

Working with other people

We understand that you may want to work with friends to complete the programming tasks. It is completely fine to collaborate with your peers at a high-level about approaches towards solving the problems. However, **the code which you write must completely be your own work and you absolutely must not copy another student's code**. We will be actively checking to ensure that you have completed the work yourself and not used another student's work – and reserve the right to ask you questions about your submission, or give you another task to complete, if we believe there has been dishonesty regarding your submission.

Marking Criteria

Item	Mark	Total
Grok Learning Modules 1-3 (18 marks – AT STANDARD)		
Module 1 – Talking to your computer	**6**	
Module 2 – Calculating things	**6**	
Module 3 – Making decisions	**6**	
Python Programming Challenges 1-3 (12 marks)		
Task 1 – Talking to your computer	**4**	
Task 2 – Calculating things	**4**	
Task 3 – Making decisions	**4**	
Grok Learning Module 4 (6 marks – ABOVE STANDARD)		
Module 4 – Manipulating strings	**6**	
Python Programming Challenge 4 (4 marks)		
Task 4 – Manipulating strings	**4**	
Grok Learning Modules 5-6 (6 marks – EXCELLING)		
Module 5 – Repeating things	**3**	
Module 6 – Storing lists of values	**3**	
Python Programming Challenges 5-6 (4 marks)		
Task 5 – Repeating things	**2**	
Task 6 – Storing lists of values	**2**	
Penalties		
Late penalty	**10% / day**	
TOTAL	**50**	

Student Outcomes

ACTDIP027	Define and decompose real-world problems	**ACTDIP030**	Implement and modify programs in a general-purpose programming language
ACTDIP029	Design algorithms represented diagrammatically and in English		