

# Year 7 Assessment Task Notification



<b>Subject:</b>	<b>Digital Technologies</b>
<b>Assessment Task:</b>	<b>Grok Learning DT Python Chatbot Project</b>
<b>Due Date:</b>	<b>Part 1 (50%): 8:45am Friday 11 May 2018 (Term 2, Week 2)</b> <b>Part 2 (50%): 8:45am Friday 15 June 2018 (Term 2, Week 7)</b>
<b>Weight:</b>	<b>100%</b>

## Context of the Task

This task allows you to demonstrate your skill in completing basic computer programming and algorithm design in the Python programming language.

## Requirements of the Task

For Part 1 (due by 8:45am on Friday 11 May 2018) you need to have completed **Project 1** in the DT Python Chatbot course on Grok Learning. There are **three** subtasks within Project 1 that need to be completed.

For Part 2 (due by 8:45am on Friday 15 June 2018) you need to have completed **Projects 2 and 3** in the DT Python Chatbot course on Grok Learning. In total, there are **five** subtasks within **Projects 2 and 3** that need to be completed.

You will be given class time to complete the other non-project modules within the DT Python Chatbot course which will provide you with the knowledge and skills to complete the projects.

## Submission Instructions

In order to submit your work you just need to **Save/Run** your code and then click **Mark** within Grok Learning. All your code submissions are timestamped and when your teacher reviews your submissions they will check that they were completed before the due date and time. You do not need to do anything else to submit your work.

## General Notes

Part 1 involves far less work than Part 2. However, we have weighted both parts the same. This is because we understand that many Year 7 students have never done programming before so we want to provide everyone with the best opportunity to get a high mark in Part 1 (being the first assessment task for this subject) which counts substantially towards your grade for the semester. Part 2 will involve more work, or a more complex nature, once everyone is familiar and more comfortable with the content.

## Marking

Grok Learning automatically marks your code by subjecting it to a battery of test cases. If it passes all the test cases, your code is marked as correct. If it fails, your code has some bugs which you need to review and fix.

Each task that you complete will be marked out of 15.

It is important that you don't hammer the auto-marker by doing something, then immediately marking your code to see whether it works. You should take the time to carefully craft your code and run your own set of tests before clicking the **Mark** button. In essence, you should only click **Mark** when you are absolutely certain that your code should work.

To emphasise this, for each of the tasks you need to complete in each project you can run the auto-marker up to (and including) **five** times without penalty. For the sixth, and every subsequent time, you run the auto-marker you will be deducted 0.5 marks.

For example, if you run the auto-marker three times and on the third attempt it passes then you would get the full 15 marks for that task. On the other hand, if you run the auto-marker ten times and it only passes on that tenth attempt you would get 12.5 marks for that task (15 marks – 2.5 marks).

**Working with other people.** We understand that you may want to work with friends to complete the projects. 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. Note, using someone else's code and changing the variable names, etc. to intentionally obfuscate the fact you have used someone else's code is still considered plagiarism.

## Outcomes/Assessment Criteria

ACTDIK024	Representation / Use of Strings Representation / Use of Integers
ACTDIP029	Design algorithms represented diagrammatically and in English, and trace algorithms to predict output for a given input and to identify errors
ACTDIP030	Implement and modify programs with user interfaces involving branching, iteration and functions in a general-purpose programming language

## Student Confirmation

By submitting your work through Grok Learning you attest that:

- This is all my own work. I have referenced any work used from other sources and have not plagiarised the work of others.

This evaluation will be completed in class following your submissions and does not need to be completed in advance. It is provided here for your reference only.

## Student Evaluation

*(Circle a number: 1 = strongly agree to 5 = strongly disagree)*

- |                                       |   |   |   |   |   |
|---------------------------------------|---|---|---|---|---|
| • This is my best work                | 1 | 2 | 3 | 4 | 5 |
| • I was organised                     | 1 | 2 | 3 | 4 | 5 |
| • I understood the task               | 1 | 2 | 3 | 4 | 5 |
| • I took notice of the marking rubric | 1 | 2 | 3 | 4 | 5 |

- I used to think and now I know:

---

---

---

- What I have learnt most about the Assessment Task:

---

---

---

- What I found most challenging about the Assessment task was:

---

---

---

- This assessment took me:      0-1      2-3      4-5      6-7      8+      hours to do

*(Circle one)*

- Next time I will:

---

---

---

- My strengths in the task were:

---

---

---