A-level induction work – Computer Science

Welcome to A-level Computer Science! At Hardenhuish we study the exam board AQA which we have taught for several years, you can find the specification here and all the resources can be found on the '***ICT Resources Notebook'*** if you want to get a flavour of some of the topics (for those of you that already have access). We understand that each year we get students with a range of skills and abilities who take the course, some of you may have studied GCSE Computing and will already have strong programming skills, whereas some of you may never have programmed before, and that's ok! The idea of this summer work is to give everyone a challenge regardless of ability – ability should just define how far I expect you to get.

What work is being set?

I am setting you all the GCSE Computing 2018 controlled assessment which asks you to create a password generator/checker. The task requires you to code and develop a system that allows a user to generate a strong password or check the strength of their current password using a scoring system.

This task is not just a programming task but also a research task, as you will be required to use the resources around you (e.g. websites, videos) to help you learn new skills and overcome problems. Independence is a core skill at A-level and this will give you a flavour of how to overcome problems using resilience and resourcefulness.

The language we are using for this task is **Python**, I know that many of you may have another preferred language of choice however this is the language you will be using for you're A-level exam so this is the one you need to practice. If you haven't downloaded Python already you can download it here. Alternatively if you cannot install Python you can use a web version which can be found here - https://repl.it/languages/python3 - I would recommend setting up an account as this will allow you to save your work.

Please note that the work you are being set is an old GCSE controlled assessment. Although I encourage you to research skills please be aware that solutions might still be online for this task and I do not expect you to copy chunks of code from working/existing solutions.

To complete this task you will need to use the following programming skills:

- Variables and printing
- Selection statements (IF statements)
- Iteration (Loops)
- Arrays
- Random function
- Functions (the solution can be solved without the use of functions)

For those of you who are already able programmers I would also like you to go one step beyond the scenario and aim to develop the solution using a GUI – for those of you who are new to GUIs I would recommend AppJar – it's really easy to set up and the website is full of help and support to get you on your way. You can download AppJar here!

My expectations for completion of work are as follows:

- Students who are new to coding should aim to complete (at a minimum) sections 1 – 3 on the question paper
- Students who have already studied computer science at GCSE are expected to try and complete all the tasks on the question paper
- Advanced programmers are expected to complete all the tasks and include a GUI

How long should I spend on the work?

You should aim to spend around 5 – 10 hours working on this project. I am unable to put expectations on what should be done when due to the following:

- Some of you will be experienced coders and will solve many of the less complex aspects of the scenario quickly, whereas some of you will need more time to focus on these
- Some of you may choose to complete the task in a different order to others (which is fine)

Due to this I will be checking in on your progress after five hours to see how you are getting on – I will not be expecting to see a working solution until the end of the 15 hours.

How should I evidence my work?

As you will all be working at different paces and solving different problems, I would like you to complete a development diary as you go. This diary needs to be completed after each hour of work where you write up and discuss:

- Where you got to during that hour (what work did you complete/challenges did you overcome)
- What current issues/problems you have that still need to be solved
- Evidence of your code at that point

This means that regardless of how far you get or how much work you complete I can still see you are making progress each hour.

Useful links:

- Python download
- Python Online
- AQA A-level Computing specification
- AppJar download (secure coders only)