

## A Level Computer Science

Welcome to A Level computer science. Over the next two years we will heavily upgrade some of your GCSE skills and learn about totally new and cutting-edge topics such as artificial intelligence and data science.

While studying the **AQA A Level (Course Code: 7157)** you will prepare for two examinations.

### **Exam 1: 40%**

An on-screen programming examination (using the Python programming language). You will be given a Skelton code at the beginning of Year 13 and will be expected to learn about it, understand it and hopefully identify some of its flaws!

During the exam you will answer general questions about the code as well as being asked to add/develop the code as well as some of your own.

### **Exam 2: 40%**

A tradition pen and paper examination which will focus on the theory content that you have learned over the two-year course. Topics will include,

- Data representation
- Computer organisation and architecture
- Computer Systems
- Consequences of uses of computing
- Networking
- Database Design and Big Data
- Object Oriented Programming and Functional Programming

### **The Project: 20%**

There is also a non-examined element to consider (20%). This is a project based on a topic that you will select. It might be something that you have taken further interest in as part of the course or outside of it. You are not restricted to Python in terms of the programming for this project. This project can be of use as a portfolio piece outside of the course if you are moving onto apprenticeships or higher education in this subject area. This is a fantastic opportunity to develop an idea over time and document your successes (and failures, as they are useful too!) and to see something through from start to finish under the scrutiny of yourself and others!

## Summer Work (extended summer this year!)

It would be fantastic if you were developing your programming and problem-solving skills over the summer.

There are a few useful resources for doing this.

1. Here you will find a tutorial on how to do just about anything you need to do with Python. It's fantastic and comes with exercises so that you can check it's all sinking in!

[https://www.w3schools.com/python/python\\_syntax.asp](https://www.w3schools.com/python/python_syntax.asp)

2. When you want a challenge that requires some problem solving, try this. They all come with working solutions, but they might not always be the same as yours as this is programming, and that's a thing!

<https://www.practicepython.org/>

3. Finally to get you in the spirit of doing a project I have created a mini project for you to try (see below)

How many days have you been alive?

**Task 0: DON'T Google ANYTHING UNTIL TASK 6!**

**Task 1:** Decomposition of the initial problem:

What are three (at least) considerations you need to make when thinking about calculating how many days you have been alive?

**Task 2:** Calculation

Calculate the number of days **you** have been alive on paper

Check your answer [https://www.1happybirthday.com/birthday\\_calculator.php](https://www.1happybirthday.com/birthday_calculator.php)

**Task 3:** Generalisation

Adjust your method so that anybody could work out how many days **they** have been alive?

Draw a flow chart of your method.

**Task 4:** Abstraction

Consider the differences between a person using your method and a computer. Explain the differences between what a human would need compared to what a computer would need to accomplish the task at hand.

**Task 5: Application**

Write the days alive program (here or transferred to here when you are done if possible <https://repl.it/>)

It should, Allow anybody to enter their birth date and correctly calculate the number of days they have been alive. You can compare your answer to the previous website.

**Task 6: Consideration**

How useable is your program? Is it robust, can you make it crash? Is it visual or text based only? How could you improve your application to make it more appealing? Try to complete some of the changes you have considered here to improve the experience of using your program.

**Task 7: Research**

Is there a different way to accomplish this programming task?

If you find out a different way, try to use it in your own program.

Explain the differences between the two methods.

Which method is more appropriate?