



Welcome to Computer Science

It's a challenging A Level course, combining Mathematics and Computing. The ability to program is important but there is an emphasis on computational thinking, as a discipline, a focus on Maths has been embedded throughout this course.

During the Summer, you could be looking at the following concepts: Binary, Logic Gates, Programming concepts and Data Structures. Revising these topics over the summer holidays would be advisable.

- Computer Science now covers a wide range of topics – not just based on programming but also the economic, social, moral and laws that relate to the discipline of Computer Science.
- Studying object-orientated concepts would be sensible, see the resources section for further details.
- Creating and executing Python programs is vital.
- Download and install Python 3.5

I look forward to teaching you all in September.

Mr Zapata

Specification: OCR A Level Computer Science:

<https://www.ocr.org.uk/qualifications/as-and-a-level/computer-science-h046-h446-from-2015/specification-at-a-glance/>

Students to join the google classroom using the following URL:

<https://classroom.google.com/r/NTM5NzI3NzAxOVpa/sort-name>

and join the class with the following code: digh2nr

Note: Upload all the completed tasks on the google classroom!

Instructions: Please select and complete two out of the four activities then submit on the google classroom above. If you don't know which activity to choose I would recommend you choose the



Activity 2 and Activity 4 as they are easy to start with. The challenges are only if you have some time and would like to do more programming during summer!

Computer Science Summer Tasks.

Activity 1

On many major roads, average speed checks are in place. Two sensors are placed a known distance apart and vehicle number plate recognition is used to identify a vehicle and the time it enters the section of road being monitored. The time is recorded when the vehicle leaves the monitored section.

By using the time taken to travel the known distance, the average speed of a vehicle can be calculated.

Write a program for calculating average speeds for a vehicle travelling through a section of road. (Speed limits for roads are 20, 30, 40, 60 and 70mph)

Activity 2

In the UK most vehicle registrations are in the format:

- two letters
- two numbers
- three letters

For example, AZ01 XYZ.

The vehicle number plate recognition system will provide this information as a string of characters. *Write a program that will determine whether a string entered meets these requirements or not and alerts the user to the correct use of the string.*

Activity 3

When a customer has a problem with a product, automated troubleshooting programs are available to help them.

If a problem with a mobile device occurs, then a troubleshooting process is invoked, with the user



responding to a series of questions that lead to a solution or advice to contact the supplier directly.

For example, a conversation might proceed as follows:

System: 'Has the phone got wet?'

User: 'No'

System: 'Has the phone been dropped?'

User: 'No'

System: 'Is the phone fully charged?' ...

Not all of the questions need to have yes or no responses but each response will need to lead to a supplementary question or a solution.

You will need to create a troubleshooting tree for a mobile device. This should be a demonstration version of the system with at around 10 possible outcomes.

Activity 4

A primary school teacher wants a computer program to test the basic arithmetic skills of her students. The program should generate a quiz consisting of a series of random questions, using in each case any two numbers and addition, subtraction and multiplication.

The system should ask the student's name, then ask 10 questions, output if the answer to each question is correct or not and produce a final score out of 10.

Further challenges

Programming Challenge 1

Average Calculator: Just to clarify, you're not creating a mediocre calculator, but a program for calculating averages.

The user should be able to enter a series of numbers, and the program should print the average of these numbers. You can use floating-point number variables, or store the input in a list.



Your program might be used to calculate average temperatures for a week, or a batting average for a cricket team, among others. You could even expand the program to print the mean, median and mode averages.

Programming Challenge 2

RGB Colour Converter: When making websites, one way of thinking about different colours is by their RGB values. RGB stands for Red, Green and Blue. Every colour can be 'made' by combining different amounts of Red, Green and Blue. Each part of the colour has a value between 0 and 255. Here are some examples:

(255, 0, 0) = RED This is a colour made up of 'full' red, and no green or blue. This makes red.

(0, 139, 139) = TEAL This is a colour made up of no red, and some green and blue. This makes teal.

Another way of representing colour is by a hexadecimal code. This is the same RGB combination, but with 2 hex digits (00 - FF) representing each of the three component colours:

#FF0000 = RED #008B8B = TEAL

Write a program to convert from a decimal RGB colour code to its hexadecimal code equivalent.

Programming Challenges 3

See this website for many more challenges <http://www.practicepython.org/>

Other Challenges

These are not related to a single programming language but are go to practice problem solving, see this link http://www.codeabbey.com/index/task_list

Resources/Books/Websites



Please buy, and aim to read as many of these as possible.

Recommended book - OCR AS and A Level Computer Science Textbook

PM Heathcote and RSU Heathcote ISBN: 978-1-910523-05-6

Computational Fairy Tales by Jeremy Kubica

But How Do It Know? by J Clark Scott

Code: The Hidden Language of Computer Hardware and Software by Charles Petzold

Websites [to learn python]

Codecademy – Python track <http://www.codecademy.com/en/tracks/python>

For students that are not confident with programming before the start of the course they could go away and work through the exercises at a website such as codecademy. Topics such as:

- Variables and string manipulation
- Conditionals and functions
- Lists and dictionaries
- Looping – Iterating over data structures –
- Bitwise operators
- Object Orientated Programming (OOP)
- File input/output

Introduction to OOP <http://www.pythonschool.net/oop/introduction-to-object-oriented-programming/>

The Hitchhikers Guide to Python – more of a reference but the tutorials are useful
<http://docs.python-guide.org/en/latest/intro/learning/>

Learn Python the Hard Way <http://learnpythonthehardway.org/book/>

note this is version 2 of Python – the school used version 3 but there is not a massive difference at this stage of learning.