AQA
Realising potential

A-LEVEL COMPUTER SCIENCE

A-level (7517)

Specifications
For teaching from September 2015 onwards

For A-level exams in May/June 2017 onwards

Version 1.5 21 January 2019

Student Handbook

Teachers:

Mr Herbert
Ms Eleftheriadi

Computer Science is a challenging A Level subject which includes a combination of theory and practical work. The best computer scientists enjoy thinking about and solving complex logical problems and puzzles: they are good at maths and physics and like to think computationally.

At A Level, students continue to use the programming constructs learned at GCSE: sequence; selection; and iteration. Writing code is an essential skill in the modern world, and at A Level, students build upon the foundations laid at GCSE and learn to program in Java, becoming more object-oriented in our approach.

The course covered is ***AQA A Level Computer Science 7517*** and is made up of two exam papers (40% each) and a practical programming project (20%).

Paper 1	Paper 2	Non-Exam Assessment (NEA)
What's assessed This paper tests a student's ability to program, as well as their theoretical knowledge of computer science from *paper 1 content* (see next page).	**What's assessed** This paper tests a student's ability to answer questions from subject *paper 2 content* (see next page).	**What's assessed** The NEA assesses student's ability to use the knowledge and skills gained through the course to solve or investigate a practical problem. Students will be expected to follow a systematic approach to problem solving.
Assessed On-screen exam: 2 hours 30 minutes 40% of A Level grade	**Assessed** On-screen exam: 2 hours 30 minutes 40% of A Level grade	**Assessed** 50-hour project 20% of A Level grade
Questions Students answer a series of short questions and write/adapt/extend programs in an electronic answer document provided by AQA. AQA will issue preliminary material, including a skeleton program (in Java) and any necessary test data.	**Questions** Compulsory short-answer and extended answer questions.	**Project documentation** including: Analysis Design Implementation Testing Evaluation Including screenshots of code and a 10 minute (maximum) video of a working system being fully tested.

Course Content

During the two-year course, students will study the following topics:

Topic	Paper	Year
Fundamentals of programming	Paper 1	
Fundamentals of data structures	Paper 1	
Fundamentals of algorithms	Paper 1	
Theory of computation	Paper 1	
Fundamentals of data representation	Paper 2	
Fundamentals of computer systems	Paper 2	
Fundamentals of computer organisation and architecture	Paper 2	
Consequences of uses of computing	Paper 2	
Fundamentals of communication and networking	Paper 2	
Fundamentals of databases	Paper 2	2nd year only
Big Data	Paper 2	2nd year only
Fundamentals of functional programming	Paper 2	2nd year only
Systematic approach to problem solving	NEA	
Non-exam assessment - the computing practical project	NEA	2nd year only

Non-Examined Assessment (NEA) Practical Programming Project

The NEA practical programming project allows students to develop practical skills in the context of solving a realistic problem or carrying out an investigation. The project is intended to be as much a learning experience as a method of assessment; students have the opportunity to work independently on a problem that interests them, over an extended period (minimum 50 hours), during which they can extend their programming skills and deepen their understanding of computer science. The most important skill that should be assessed through the project is an ability to create a programmed solution to a problem or investigation.

The NEA practical programming project is a 50-hour project which is expected to be completed outside of the classroom. Students are strongly encouraged to begin their NEA projects at the end of Year 12 (summer term), ***before*** the start of Year 13.

The NEA project must be a coded solution or investigation of significant size and complexity that meets the requirements of a real-world problem (e.g. a website with a database back-end, a networked game, a smart phone application, an investigation in an area of machine learning). All projects must be discussed and agreed by the class teachers beforehand.

Deadlines and milestones will be set throughout the second year to ensure that students stay on track. All NEA work must be complete two weeks prior to the Easter break of the second year. *It is an expectation that students keep multiple backup copies of their coursework in case of accidental mishap.*

Students are required to follow a systematic approach to problem solving and complete a project report with the following sections: *analysis; design; implementation; testing; evaluation*. Students are to screenshot all their code and make a short video (10mins max) of their working program.

The NEA project is worth 20% of the final A Level grade.

Examined Assessment

Paper 1 (the *Programming Paper*) is a two and half hour exam to be completed on a computer.

Several weeks prior to the first exam, a program (the skeleton code) is released that forms part of the *Paper 1* exam. This code will be analysed in class during the build up to the exam. Some of the tasks set in the exam will require students to identify sections of code to demonstrate their understanding and knowledge of programming, as well as expecting them to write new code or fix errors within the program. Other sections of the exam will require that students answer a series of short questions and write/adapt/extend programs in an electronic answer document, based on the topics listed in the table above.

Paper 1 is worth 40% of the final A Level grade.

Paper 2 (the *Theory Paper*) is a two and half hour standard exam to be completed on paper. Students will be required to answer short-answer and extended answer questions based on the topics listed in the table above.

Paper 2 is worth 40% of the final A Level grade.

Programming

A fundamental part of computer science is programming. As with any natural "spoken" language, to become really good in it (fluent), students must practice frequently, and a lot! If students only write code during lesson time, and do not practice it on their own, in their own time, then they will really struggle to keep up. At A Level, students write code in Java.

The Java tutorials available on YouTube created by *TheNewBoston* are highly recommended. These tutorials are short (5-10mins max) and are easy to follow with narrated steps to guide you through each section.

The direct link to the Java tutorial playlist is:
https://www.youtube.com/watch?v=HI-zzrqQoSE&list=PLFE2CE09D83EE3E28

If you have any difficulty accessing this, please speak with your class teacher for advice.

There are also plenty of other online resources to help you become a better Java programmer.

Microsoft OneNote

It is advisable that you write extensive notes during class in a traditional notebook, and then type these up later using your online *Class Notebook*. Whenever we discuss a topic in class, you should be going home that same day (whilst the knowledge is still fresh) and reading further around the topic (or researching online) to embed this information, and update your notes further.

You must read through your notes regularly and revisit topics to keep your understanding fresh in your mind. The best students always practise exam questions from past papers, which have been made available on your class' Microsoft Team.

Reading

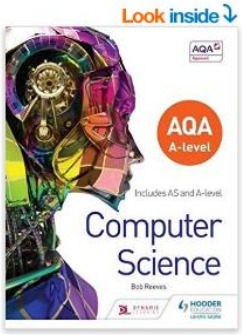
Reading is essential. If you do not read, then you are already putting yourself at a massive disadvantage. Students who read regularly and read around the topics covered in class, are much more likely to achieve higher grades.

All these books can be found on ***Amazon***, and occasionally for half the price on ***eBay***.

Whilst I respect the environment, it is always recommended to have a hard copy of a textbook (as opposed to an electronic copy or PDF). This way you can always step away from your computer, get comfortable, and **READ**.

Essential Reading

AQA A level Computer Science, Bob Reeves
The Pattern on the Stone: The Simple Ideas That Make Computers Work, Daniel Hillis

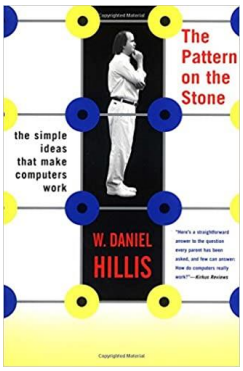


Desired Reading

AQA AS and A Level Computer Science, PM Heathcote
A Level Computer Science for AQA Unit 1, Kevin Bond
A Level Computer Science for AQA Unit 2, Kevin Bond

Above and Beyond Reading

Tackling A Level Projects in Computer Science AQA 7517, PG Online



Documentaries

There is a lot of stuff on technology out there, and this list is constantly growing. I have really enjoyed many of these.

On Netflix:
The Great Hack (documentary about the Cambridge Analytica scandal)
Inside Bill's Brain (documentary on Bill Gates)
The Social Network (drama on how Facebook was created)

Elsewhere:
Lo and Behold (documentary on how the Internet has changed the world)
Silicon Cowboys (documentary about Compaq vs IBM)
Brexit: the Uncivil War (drama about Cambridge Analytica)
Imitation Game (film drama about the great Alan Turing)

For the module *Consequences of Uses of Computing* you must have a good grasp of the ethical, legal, cultural, and environmental impact of technology (both positive and negative), and all the issues that surround it.

Watching documentaries and the news to keep up with the constant changes in technology, the influence technology has on our lives and society, as well as the laws that govern and control how technology is used responsibly is crucial.

Use the *BBC Technology* section as a good starting point for this:
https://www.bbc.co.uk/news/technology

Our Expectations

Full attendance *
Be an active participant in class, group projects, and any set challenges
Listen attentively in lessons
Take detailed notes in lessons as instructed
Use a text book to reinforce and embed understanding
Use online resources to research topics
Ask when unsure of anything
Practice coding regularly and experiment with code outside of the classroom
Complete all home-learning tasks
Meet coursework deadlines

* *If a lesson is missed a student must email the class teacher to find out what was missed. Students must then study the missed content before asking the teacher to clarify anything not understood.*

What you can expect

Well-prepared lessons
Interesting and exciting activities
Resources for each topic
Help with areas not understood
Verbal and written feedback
Access to past exam papers
Model answers
Project and group work
Relevant experience for a tech jobs or university degree