










Contents:

	Page
 Course/specification overview	3
 Our department expectations	5
 Review/revise	6
 Watch	7
 Listen to	8
 Read	9
 Research	10
 Complete	11
 Appendices/resources	12



Course/specification overview

The course specification can be viewed [here](#)

Aims and learning outcomes

The aims of this qualification are to enable learners to develop:

- an understanding of and ability to apply the fundamental principles and concepts of computer science including; abstraction, decomposition, logic, algorithms and data representation
- the ability to analyse problems in computational terms through practical experience of solving such problems including writing programs to do so
- the capacity for thinking creatively, innovatively, analytically, logically and critically
- the capacity to see relationships between different aspects of computer science
- mathematical skills
- the ability to articulate the individual moral, social, ethical, legal and cultural opportunities and risks of digital technology.

Key features of this specification

The OCR A Level in Computer Science will encourage learners to be inspired, motivated and challenged by following a broad, coherent, practical, satisfying and worthwhile course of study. It will provide insight into and experience of how computer science works, stimulating learners' curiosity and encouraging them to engage with computer science in their everyday lives and to make informed choices about further study or career choices.

The key features of this specification encourage:

- emphasis on problem solving using computers
- emphasis on computer programming and algorithms
- emphasis on the mathematical skills used to express computational laws and processes, e.g. Boolean algebra/logic and comparison of the complexity of algorithms

Centres and learners have the opportunity to:

- produce a coding focussed programming project
- choose the project title and problem to be solved
- choose any suitable programming language

Content Overview	Assessment Overview	
<ul style="list-style-type: none"> • The characteristics of contemporary processors, input, output and storage devices • Software and software development • Exchanging data • Data types, data structures and algorithms • Legal, moral, cultural and ethical issues • Elements of computational thinking • Problem solving and programming • Algorithms to solve problems and standard algorithms <p><i>The learner will choose a computing problem to work through according to the guidance in the specification.</i></p> <ul style="list-style-type: none"> • Analysis of the problem • Design of the solution • Developing the solution • Evaluation 	<p>Computer systems (01)</p> <p>140 marks</p> <p>2 hours and 30 minutes</p> <p>written paper</p> <p>(no calculators allowed)</p>	<p>40%</p> <p>of total</p> <p>A level</p>
	<p>Algorithms and programming (02*)</p> <p>140 marks</p> <p>2 hours and 30 minutes</p> <p>written paper</p> <p>(no calculators allowed)</p>	
	<p>Programming project 03* – Repository or 04* – Postal or 80 – Carry forward (2018 onwards)*</p> <p>70 marks</p> <p>Non-exam assessment</p>	<p>20%</p> <p>of total</p> <p>A level</p>



Our department expectations

The Computer Science department has high expectations of students. We expect you to be engaged and willing to learn for yourself, be respectful to others in your classes and make your very best efforts in all lessons and homework. All homework is to be submitted by the deadline stated and in the required format. In return your teachers will provide you with regular feedback to enable you to progress.

Lesson Preparation and Organisation

- Pre-reading from textbook or online when requested.
- Organisation: textbook, folder and pencil case in every lesson.
- Regularly check your email and Satchel1

(please note calculators are **not** required in lessons as not permitted in exams)

Independent Study

- Catch up on missed work due to absences
- Use non-contact study periods (timetabled) for pre-reading, structured reviewing of learned material and practical work as required
- Revision for end of unit tests and exams
- Use study skills and revision skills that have been taught to you
- If below target grade, must attend study sessions for support with teacher, complete practise questions on topics taught (to check understanding)



Review/revise

Revision

Please use the following link to help inspire creativity into your revision techniques:

<https://hayestl.com/students/>

Coding

It is assumed that you will have a reasonable level of proficiency in python coding before starting this course in September. If you have not coded in python before or feel you need to brush up your python skills please use the resources section to help.



Watch

This talk on the impact that artificial intelligence and robotics will have on jobs in the future.

[The jobs we'll lose to machines - and the ones we won't](#)

TED Talk – Anthony Goldbloom

This talk on the internet of things [Everything around you can become a computer](#)

TED Talk – Ivan Poupyrev

This talk which explores some of the philosophical ideas surrounding artificial intelligence.

[What happens when our computers are smarter than we are?](#)

TED Talk – Nick Bostrom

Films:

[Hackers](#): The hero of the story is arrested for writing a computer virus as a teen, and as an adult later works with his friends to take down a plot to release a dangerous computer virus.

[Wargames](#): Oops! An unsuspecting computer prodigy accidentally starts World War III when he hacks into a military computer to play a war game. Enjoy the ideas of computer control, automation, and the importance of being error free.

[The Matrix](#): Most computer programmers live a pretty typical life, but Neo finds out that his reality is in fact not a reality at all, and rather a computerized artificial world. Neo offers inspiration to all who care about the importance of real human life over virtual reality.

[Tron](#): Have you ever felt trapped by your computer? In this movie, a hacker actually is, and has to overcome a program that holds him captive. Students can enjoy this movie, and it's sequel Tron: Legacy for their lesson in being bigger than your computer program.

[Antitrust](#): If you think you're set for life when you get hired for your programming dream job, think again. This movie takes a look at the potential for ruthlessness and danger in the computer science industry.

[Ghost in the Shell](#): An animated movie from Japan, Ghost in the Shell tells the story of a well-connected future set in 2029 full of hackers and the cyborgs that prevent them. Ghost in the Shell will teach computer science students to be wary of over connectedness.

[The Imitation Game](#): Alan Turing coding our way to victory in WWII



Listen to

This radio interview with software pioneer Dame Stephanie Shirley.

BBC Sounds [Life Scientific](#)

This radio programme which explores whether computers can beat humans at games.

BBC Sounds [Seriously...Game over humans](#)



Read

Background reading

Code: The Hidden Language of Computer Hardware and Software

By Charles Petzold

The Soul of A New Machine

By Tracy Kidder

The Second Machine Age: Work, Progress, and Prosperity in a time of brilliant technologies

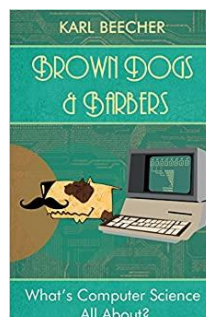
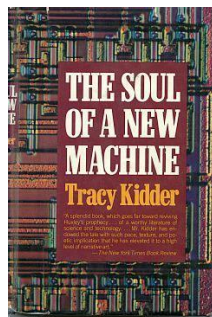
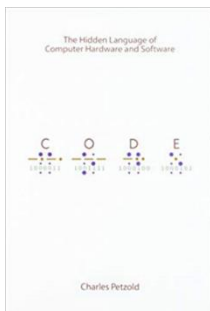
By Erik Brynjolfsson & Andrew McAfee

Brown Dogs and Barbers What's Computer Science All About?

By Karl Beecher

Ready Player One

Ernest Cline



Read this students or not – friends or disagree!

article about whether should learn how to code start a debate with about whether you agree

[Teaching waste of time](#) The Telegraph

[children coding is a](#)

this article about the future of teaching in the world of AI.

[Intelligent machines will replace teachers in ten years](#) The Independent



Research

Programming languages

Investigate OOP, C++ and Visual Basic.Net

Your Ted Talk

Do any research necessary for your Ted Talk (See the complete section number 3)



Complete

1. This course which explores how the way we work will change in a world of automation.

[The future of work - preparing for disruption](#) *Open Learning Campus*

2. This course which is ideal for those who would like to set up a coding club for younger students when school is back open.

[Prepare to run a coding club](#)

We have 10 Pis at school and are looking for year 12s to run an after school club

3. Make a "TED" talk about **any** Computer Science topic that interests you
Eg:

- the inside of a computer
- how computer Science affects my life in lockdown
- why certain games are addictive
- how studying computer science helped me from your future self
- how my gaming consoles have changed through the years

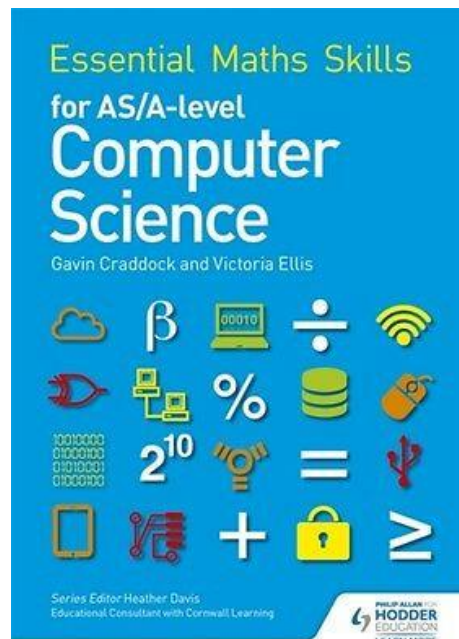
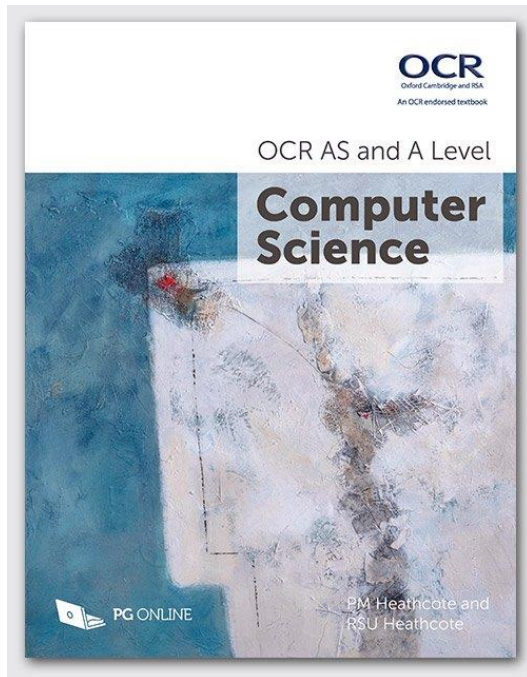
These must be ready to present to the class in the second week of the first term in year 12.

4. Try coding in one of the languages that you researched in the previous section and find out how the syntax is different from python



Resources

Textbooks



**Craig n
Dave**

[youtube](#)
short

introductory videos for most topics in the

specification

Teach-ict

[Online](#) resources for learning most topics in the specification

Python

[Snakify](#) tutorials and challenges for python coding

<https://www.python.org/about/gettingstarted/>

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