

Year 11 to Year 12 Bridging Work

|  |  |
| --- | --- |
| Subject | Computer Science |
| Course | A-Level |
| Awarding Body | OCR |

**Course/specification overview**

The course specification can be viewed [here](https://www.ocr.org.uk/Images/170844-specification-accredited-a-level-gce-computer-science-h446.pdf)

**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 computational thinking skills

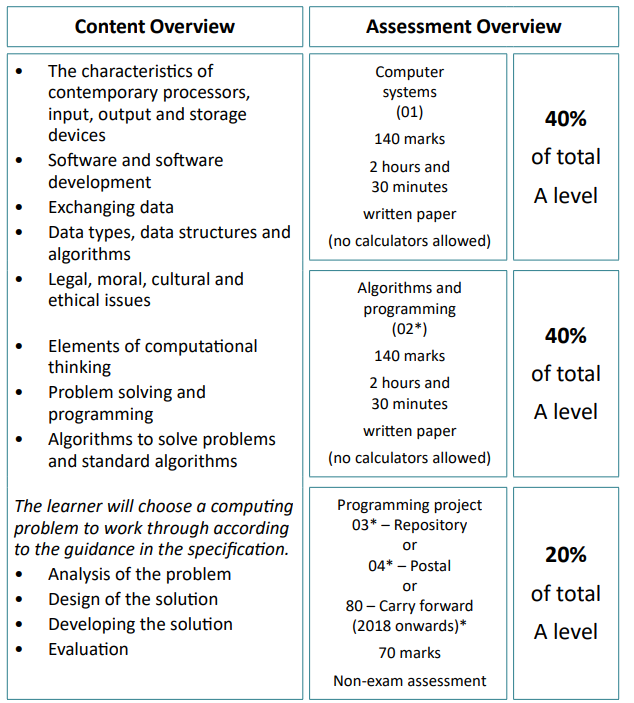
• Emphasis on computer programming and algorithms

• Emphasis on the mathematical skills used to express computational laws and

processes, e.g. Boolean algebra/logic, comparison of the complexity of algorithms and floating-point arithmetic

**Learners will:**

• Complete a programming project for the Non-Examination Component of the subject. They will produce a programmed solution to solve a problem. The project will include these stages: Analysis, Design, Implementation, Testing and Evaluation



**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. You are expected to be proactive, engaging in content outside of the lesson and developing programming skills independently.

**Lesson Preparation and Organisation**

• Pre-reading from Textbooks, school resources on SharePoint and Online resources as needed.

• Organisation: Workbooks, folder and pencil case in every lesson.

• Regularly check your email and MS Teams

**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 practice questions on topics taught (to check understanding)

**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 done any programming before or feel you need to brush up your Python skills; you **mus**t start now to familiarise yourself with the programming language – see resources suggested below.

For the NEA component, you can learn other programming languages, independently, for your project – see the Exam board specification for programs allowed. You must consider how long this will take on your part.

**Watch the following Talks**

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](https://www.ted.com/talks/anthony_goldbloom_the_jobs_we_ll_lose_to_machines_and_the_ones_we_won_t?referrer=playlist-what_happens_when_the_robots_take_our_jobs#t-3590)

TED Talk – Anthony Goldbloom

This talk on the internet of things

[Everything around you can become a computer](https://www.ted.com/talks/ivan_poupyrev_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?](https://www.ted.com/talks/nick_bostrom_what_happens_when_our_computers_get_smarter_than_we_are?referrer=playlist-what_happens_when_the_robots_take_our_jobs)

TED Talk – Nick Bostrom

**Films to watch**

**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

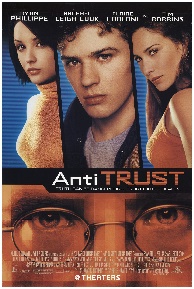
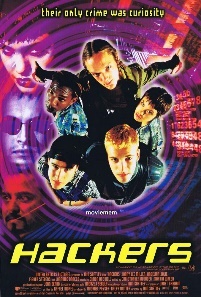
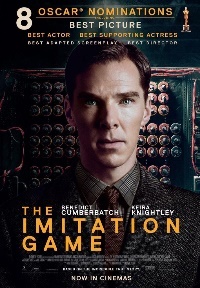
computer science industry.

**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



**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



**Resources**

**Python**

[Snakify](https://snakify.org/en/) tutorials and challenges for python coding

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

<https://www.csnewbs.com/python>

**\*\*\*You can download Python onto your computer from:**

https:// [www.python.org/downloads/](https://www.python.org/downloads/)

**There are also lots of Online compilers where you can code on the go like:**

https://www. programiz.com/python-programming/online-compiler/

https://www.w3schools.com/python/python\_compiler.asp

**Complete the following**

**Task 1**

In this task you get to investigate any area of emerging computer technology which interests you.

You can pick any area which interests you, but examples could be:

• Artificial intelligence

• Robotics

• Automated self-driving cards

• Quantum computing

Watch the videos on Ethical, morale and cultural issues in ICT and Computing.

There are 5 videos to watch.

https://student.craigndave.org/videos/slr-17-ethical-moral-and-cultural-issues

**In no more than ONE side of A4 summarise** the area you have chosen under the following four headings:

1. What is the area?

2. What are the possible Social, Moral, Cultural and Ethical benefits of this technology on

society

3. What are the possible Social, Moral, Cultural and Ethical risks of this technology on society

4. My conclusion on this technology and what it will mean for our world 10 years from now

**Task 2 - Thinking Abstractly**

**Activity 1**



Colour the map above, making sure that countries that share a border are not the same colour. You should also try and use as few colours as possible.

How many colours did you need?

* How can you be sure?

Extension – adding rules (‘constraints’)

Countries 4 and 5 have now become one country. Does this change the graph and the colouring?

Country 3 does not exist anymore, and a new constraint has been introduced, that you must only have two colours. Is this possible?

**Activity 2**

8 kinds of animal, some eat others:

1 - Lion eats Baboon, Hyena, Zebra and Meerkat

2 - Baboon eats Meerkat

3 - Grasshopper

4 - Zebra

5 - Crocodile eats Grasshopper and Zebra

6 - Hyena eats Zebra, Baboon, Meerkat and Grasshopper

7 - Meerkat eats Grasshopper

8 - Cheetah eats Zebra

How would I work out a safe set of zoo allocations, so no animal eats another?

**Are there any problems that you can see with the model?**

Extensions

Construct your own graph of a different food web. Fish or birds, for instance.

You could also draw your own ‘country’ like in activity 1 and draw a graph based on it.

**Task 3 - Data types, data structures and algorithms**

Activity 1

Converting between denary, binary and hex

| No. | Denary | Binary | Hex | Binary value plus 00011110 |
| --- | --- | --- | --- | --- |
| 1 | 1 |  |  |  |
| 2 | 5 |  |  |  |
| 3 | 10 |  |  |  |
| 4 | 22 |  |  |  |
| 5 | 40 |  |  |  |
| 6 | 77 |  |  |  |
| 7 | 91 |  |  |  |
| 8 | 121 |  |  |  |
| 9 | 144 |  |  |  |
| 10 | 168 |  |  |  |
| 11 | 170 |  |  |  |
| 12 | 200 |  |  |  |
| 13 | 211 |  |  |  |

**Task 4 - Python programming tasks**

You have two tasks to complete. Write and save each program with a suitable name, test each program thoroughly to make sure it works, add comments to explain your code. The tasks provide opportunities to demonstrate a range of different programming skills.

**Activity 1**

Create a program that will allow the user to specify a range of numbers (a minimum and a maximum) and then the computer selects a random number between those two numbers.

The user must try to guess the number that the computer has selected and the computer states if they have guessed correctly. If the user has not guessed correctly the computer must say if they are more or less than 10 away from the number it had selected. This process is repeated until the user guesses the number correctly and a suitable message is displayed telling the user how many guesses they took to guess the number correctly.

The user should then have the option of playing again or quitting the game.

Activity 2

A small local estate agent wants you to write a program to help them work out the floor size of a property. Your program must ask the user how many rooms the property has and then for each room it should ask for the width and length of the room. You can assume all rooms are rectangular and there are no unusual shaped rooms. Your program must display the total area of the floor space in a user-friendly format.

**Useful link to learn python:**

<https://www.codecademy.com/catalog/language/python>

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

<https://www.csnewbs.com/python>