Warlingham
Sixth Form College

Year 11 > Year 12 Transition
Summer Term
A Level Computer Science

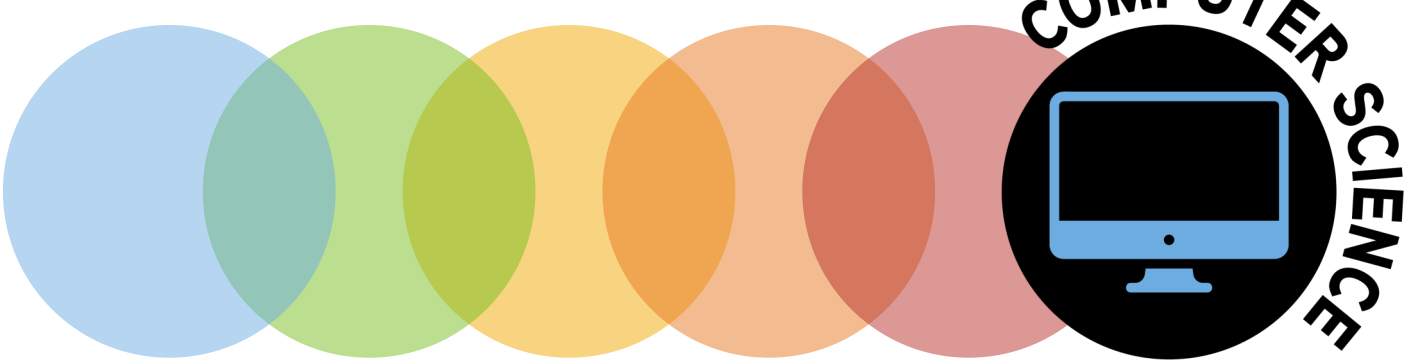


TABLE OF CONTENTS

COURSE OVERVIEW

You have made an excellent choice in deciding to study Computer Science at A Level.

Computer Science is a practical subject where students can apply the academic principles learned in the classroom to real-world systems. The aims of this qualification are to enable learners to develop the following:

- An understanding and ability to apply the fundamental principles and concepts of computer science, including: **computational thinking** (abstraction, decomposition, pattern recognition) **logic and Boolean logic, algorithms, data representation and object oriented programming (OOP).**
- The ability to analyse problems in computational terms through practical experience of solving such problems, including writing programs (using python or java programming languages).
- The capacity to think creatively, analytically, logically and critically.
- Mathematical skills to understand data types and primitive data types. The ability to solve binary arithmetic and floating point arithmetic and solve logic problems using Boolean algebra.

This pack contains a programme of activities and resources to prepare you to start an A Level in Computer Science in September. It is aimed to be used throughout the remainder of the summer term and over the summer holidays to ensure you are ready to start your course in September.

The pack is divided into some of the key topics you will study in A level Computer Science. There are a range of different activities to do in each topic area. The world of computer science continues to develop at an amazing rate. The challenge for you as a computer scientist is to be able to respond to this everchanging world and to develop the knowledge and skills that will help you to understand technology that hasn't yet been invented!

COURSE OVERVIEW

OCR A Level Specification

Students must take all three components to be awarded the OCR A Level in Computer Science.

Component 1—Computers Systems (01) Written exam paper (2hrs 30mins) worth 40% of total A Level

* The characteristics of contemporary processors, input, output and storage devices.
* Types of software (operating systems) and the different methodologies used to develop software.
* Data exchange, database concepts, introduction to SQL.
* Networks, network security and threats, HTML and web technologies.
* Data types, Boolean algebra, data structures and algorithms (analysis and design of algorithms).
* Legal, moral, cultural and ethical issues.

Component 2—Algorithms and programming (02) Written exam paper (2 hrs 30mins) worth 40% of total A Level.

* What is meant by computational thinking (thinking abstractly, thinking ahead, thinking procedurally, etc.)
* Problem solving and programming—how computers and programs can be used to solve problems.
* Algorithms and how they can be used to describe and solve problems.

Component 3—Programming project (03) - Non-exam assessment (NEA) worth 20% of total A Level.

* Students are expected to apply the principles of computational thinking to a practical coding programming project.
* The project is designed to be independently chosen by the student and provides them with the flexibility to investigate projects within the diverse field of computer science.
* Students are expected to analyse, design, develop, test, evaluate and document a program written in a suitable programming language.

COURSE OVERVIEW

The key features of this specification encourage the following:

- 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.
- Theory of computation.
- Fundamentals of data representation.
- Fundamentals of computer organization and architecture.
- Systematic approach to problem solving.
- Learn about data structures. Data in a computer program is organized using a data structure. There are different methods for organizing data. Arrays are a common tool used to organize data when programming.
- Understand databases and how data is exchanged between different systems.
- Legal, moral, cultural and ethical issues surrounding the use of computers and ethical issues that can or may in the future arise from the use of computers.
- The use of algorithms to describe problems and the measures and methods to determine the efficiency of different algorithms, Big O notation.
- Develop and complete a programming project, choosing project title and problem to be solved and using the agile approach (*agile software development is a process for developing software, allowing you (the developer) to create your software program in the way most suited).*

OUR EXPECTATIONS

College Expectations for Academic Success

The College will work closely with all students and parents to create a purposeful, creative and stimulating environment in which students are encouraged to fully develop - both academically and personally.

We will expect you to take responsibility for your own behaviour and learning. The current College Committee along with the student body have discussed and agreed that students should commit to:

- Ensuring academic success through regular attendance and punctuality at all required registrations, lessons, supervised study lessons and Inspire Periods. Attendance which drops below 95% reduces Key Stage 5 performance by at least one grade, so it is taken very seriously.
- Completing all set tasks on time to the best of your ability, making full use of study periods and homework to enable you to meet all deadlines.
- Using study time effectively by bringing all required equipment and resources with you and making full and regular use of the College study rooms and LRC, respecting the need for silent studying conditions.
- Working closely with all your teachers to develop an effective working relationship based on mutual respect and discussing your work with them on a regular basis and meeting targets set.
- Developing your skills as an independent, self-evaluative learner and work closely with your tutor in monitoring and discussing your academic progress. As an independent learner, if you miss a lesson, it is your own responsibility to find the teacher and catch up with the work missed.
- Organising your work efficiently and effectively into folders for each subject, making full use of individual subject expectations and using Cornell Notes daily to ensure work in your folders is relevant and meaningful.
- Keeping mobile phones out-of-sight in all classrooms and during assemblies so that lessons are not disturbed and/or important information is missed.
- Attending all parents' evenings and arrange appointments with your teachers to discuss your progress and work.

OUR EXPECTATIONS

Course-specific Expectations for Academic Success

We will expect students:

- to have a technical interest in computing and computer programming.
- to stay up to date with current technology news and developments in computing.
- to develop your computer programming skills outside of the classroom to ensure you can program with confidence.
- to engage in classroom discussion and contribute to the learning environment and use free independent study periods to review and recap your learning.
- to be a problem solver…..if you are stuck then make an effort to find out the answer for yourself. Use Google, it contains so many fantastic tutorials and guides to help you. Reading a textbook/revision guide related to the topic.
- to be proactive and actively look for ways to expand your knowledge and get better at each stage and topic.

USING CORNELL NOTES

The Cornell Notes system is a note-taking system devised by Walter Pauk, an education professor at Cornell University. It is a proven method that establishes a more effective learning process.

It is designed to help the user think and reflect upon the notes they have made as well as making them more useful for revision purposes.

Please click here to watch a video that explains how to take Cornell Notes properly.

A Level Computer Science students are expected to become more independent with their learning.

Good note taking and presentation of work is an essential skill that is needed for success on an A Level course. Poorly written notes reveals a lack of understanding and do not provide any chance for revision and review at a later date.

We therefore recommend the Cornell note taking method as an effective way of taking notes.

Top Quality Notes include:

- Use of different colours to highlight key terms, key information and headings.
- Use of callouts and underlining to highlight key points and terms.
- Good use of white space to break up blocks on information.

Title of Chapter in Textbook or of PowerPoint

Reduce & then Recite - Create questions which elicit critical thinking, not 1 word answers - Write questions directly across from the answers in your notes. - Leave a space or draw a line separating questions.	**Record for Review** - Write headings and key words in different colours - Take sufficient notes - Skip a line between ideas and topics - Used bullet points and abbreviations - Include diagrams/drawings for clarification
Reflect & Recapitulate *(summarise the main points)* In your own words and using complete sentences, write a sentence summary paragraph. Your summary should cover the main concepts of the notes, be accurate and have adequate details.	

Use the Cornell notes document template. (Screen shot on the left.)

REVIEW / REVISE

Computational thinking (Task time approximately 45-60mins)

Watch this video:

https://youtu.be/euFj8D1A1Kw **and make notes using the Cornell Notes document.**

Read the following:

https://www.cambridgemaths.org/Images/espresso_19_computational_thinking_in_the_classroom.pdf

Answer the following using complete sentences and with examples:

1. What is an algorithm?
2. How can you create an algorithm? And provide an example.
3. What is computational thinking?
4. Why is computational thinking required in the study of computer science?
5. Define the term 'abstraction' and provide an example.
6. What is decomposition and provide an example.

So what actually is 'computational thinking'? You should have a basic understanding from your GCSE years**.** It is the thought processes involved in problem solving, so that the solutions are represented in a form that can be effectively carried out by an information-processing agent, such as a computer. Core concepts involved in computational thinking include: algorithmic thinking - developing a set of instructions or sequence of steps to solve a problem; evaluation - ensuring a solution is fit-for-purpose; decomposition - breaking a problem down into its component parts; abstraction - hiding detail or removing complexity without losing the important detail required to solve a problem; generalisation - finding a general approach to a set of problems.

Creativity is important when applying computational thinking principles to a problem. Programming is a fundamentally creative skill - whether it is used to create a search algorithm, build an app or design a website.

Why is it important in the workplace? So much of modern day business is about problem solving - whether that's making small improvements to enhance the efficiency of a business, or creating breakthrough products and services for consumers. Computational thinking runs through every aspect and function of a modern business. It has become more crucial in the 21st century workplace where so much is now data-driven - analysing consumer behaviour, the movement in financial markets and the performance of public services, like health or policing, are just a few job roles that require individuals to be able to think through problems in a way that a computer could understand.

WATCH

BBC Click

BBC Click is the programme for everyone interested in the internet and computing.
Whether it's e-commerce, new developments and products, or gadgets and games, BBC Click looks at the tools that will revolutionise business and personal life in the future.

This is a really useful website for staying up to date with current technological news and support you in the A level topic concerning ethics, morals, legal and cultural issues surrounding computer science.

Task 1: BBC click review of 2020
BBC click meets many-armed robot on wheels that's making keyhole surgery more available.
And we see one of the world's first trials of an artificial intelligence algorithm to help doctors quickly distinguish between Covid-19 and cancer treatment toxicity on scans. These decisions could be a matter of life and death and are challenging for the human eye to detect.

Watch the video:
https://www.bbc.co.uk/iplayer/episode/m000pc4w/click-07112020

Write an article, summarising one of the topics from the Click review, highlighting one of the aspects they discussed and what were the advantages, or disadvantages of this.

Task 2: Artificial Intelligence (AI) and Biases
We explore how biases built into everyday technologies impact people's lives and how the tech industry is trying to address it.

Watch the video:
https://www.bbc.co.uk/programmes/m000vl08

- You will learn about AI at A Level and will be required to answer an essay question in the exam that may relate to AI.
- Do some further research on AI and see below example exam question and have a go at answering.

Answer the following question:

"Developments in Artificial Intelligence mean that in twenty years time most people will be unemployed." Discuss whether or not you agree with this statement.

LISTEN TO

Network security and threats

You will discuss and learn about systems security, firewalls and encryption during your A level.

Listen to the following Podcast which talks about security, hacking and university hacking.

Podcast: https://www.smashingsecurity.com/176

Task 1: Write a summary from the podcast—what are the key points they discuss? *They talk about some Zoom meeting 'hacks', the Computer Misuse Act, Cybercrime, data breaches at Warwick University and lots more!*

Task 2. Watch/listen to the following Ted Talk: "Why study computer science!"

https://youtu.be/t3Y4p-6YWnQ

Answer the following:

1. Why have you chosen to study Computer Science at A Level?
2. What do you aim to achieve from the course?
3. What skills are you aiming to develop?
4. What other A Levels have you chosen and why?
5. What programming languages have you used?

READ

Data representation

During your GCSE you learnt about binary, denary, hexadecimal, binary arithmetic and ASCII. For A level you will also learn about the following; (some of this is covered in year 2, so don't worry you don't need to know it all now!).

* representing negative numbers using two's complement
* fixed point binary numbers
* floating point arithmetic

Complete the following tasks and then complete the worksheet/test to help remind you of the skills from GCSE or re-learn the basic data representation techniques.

1. Read the following BBC bitesize data representation pages (10 pages)
   https://www.bbc.co.uk/bitesize/guides/zfspfcw/revision/1
2. Complete the quiz and make a note of your score:
   https://www.bbc.co.uk/bitesize/guides/zfspfcw/test
3. Read the following on data representation
   https://www.bbc.co.uk/bitesize/guides/zsnbr82/revision/1

Systems architecture

One of the first topics you will start at A Level is about the components of a computer. The CPU, also known simply as the processor, has a number of different components which enable it to carry out its task of executing instructions. These components include:

- Control unit (CU)
- System Busses (Control, Data & Address bus)
- ALU (Arithmetic Logic Unit)
- Dedicated registers
  - ⇒ Program counter (PC)
  - ⇒ Current instruction register (CIR)
  - ⇒ Memory address register (MAR)
  - ⇒ Memory data register (MDR)
  - ⇒ Accumulator (ACC)

The sequencing of these instructions can be divided into 3 phrases using the :

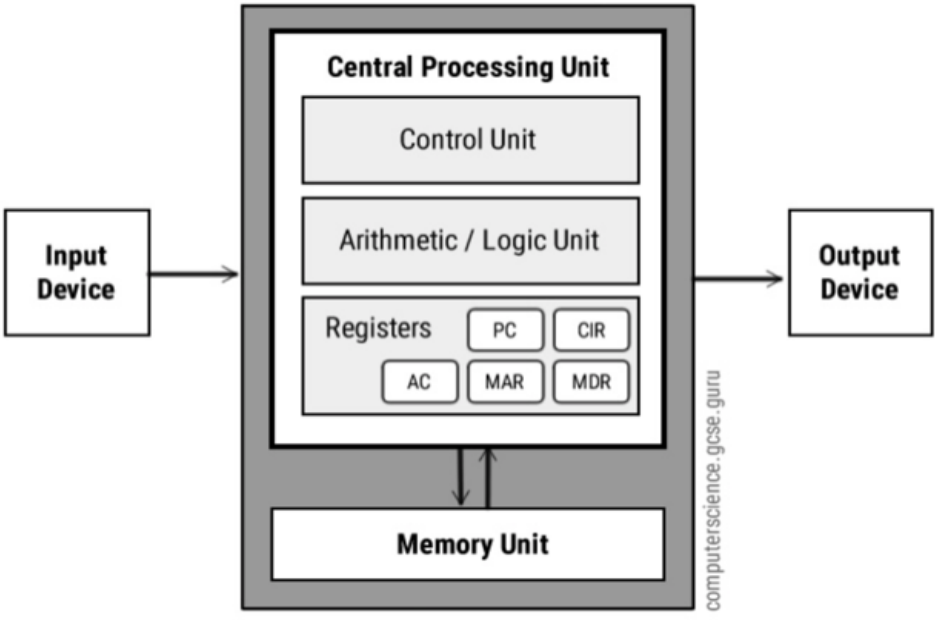
* **Fetch—Decode—Execute** cycle.

Tasks:

1. Read about the history of computing. Choosing at least 6 historic computing moments, create a time-line explaining each of these and the impact it has had on computing today.

https://www.computerhistory.org/timeline/computers/

2. Research each of the keywords above (components and registers) and **create a revision guide or poster,** which explains each register and component. You can use the diagram (or create your own) below to complete this task.



COMPLETE

Python Programming and OOP (Object Oriented Programming)

Python programming (or Java if you would prefer to use) is fundamental for A Level computing. You will be required to undertake the project using a programming language and this will require a good understanding and ability to develop a program and problem solve your programs successfully.

* You will need to be programming regularly during this course (weekly) during your own time and during free study periods.

Course 1: Python programming

Complete the following python programming course—this will take you approximately 10-15 hours *(you will get a certificate at the end to share with your teacher)*

https://www.futurelearn.com/courses/block-to-text-based-programming

Course 2: OOP (Object Oriented programming)

Complete the following OOP Python course, which will take you 10-20 hours. OOP is fundamental to A level and will help with your programming project. OOP is how the world is viewed as a collection of objects. An object might be a person, car or animal. You can then make 'blueprints' with that object and re-use its attributes and behaviours.

What is Procedural programming (Python) vs OOP (Object Oriented Programming)

Procedural Programming can be defined as a programming model which is derived from structured programming, based upon the concept of calling procedure. Procedures, also known as routines, subroutines or functions, simply consist of a series of computational steps to be carried out. During a program's execution, any given procedure might be called at any point, including by other procedures or itself.

Object Oriented Programming can be defined as a programming model which is based upon the concept of objects. Objects contain data in the form of attributes and code in the form of methods. In object oriented programming, computer programs are designed using the concept of objects that interact with real world. Object oriented programming languages are various but the most popular ones are class-based, meaning that objects are instances of classes, which also determine their types.

1. Watch this video about **OOP**: https://youtu.be/pTB0EiLXUC8

2. Go to the following **OOP course** to complete: *(you will get a certificate at the end to share with your teacher)*

https://www.futurelearn.com/courses/object-oriented-principles

APPENDICES / RESOURCES

Read/learn—You can use the full BBC Bitesize OCR GCSE course (read and do the tests) to recap your knowledge to prepare you for A level computer science.

https://www.bbc.co.uk/bitesize/examspecs/zmtchbk

Watch the Alan Turing movie—The Imitation Game. Alan Turing cracked codes produced by the German military's seemingly unbreakable Enigma machine during World War II using math, engineering and still-to-be-invented computer science.

https://www.rottentomatoes.com/m/the_imitation_game

Read/puzzles —CS4FN (Computer Science for Fun) is a magazine on computer science aimed at school students "Explore how computer science is also about people, solving puzzles, creativity, changing the future and, most of all, having fun." It is printed twice a year and has an associated website with additional articles.

News—Wired WIRED IS WHERE tomorrow is realised. It is the essential source of information and ideas that make sense of a world in constant transformation. The WIRED conversation illuminates how technology is changing every aspect of our lives—from culture to business, science to design.

https://www.wired.com/tag/computer-science/

Programming—SoloLearn Python and Java—great free website to improve your skills and knowledge. https://www.sololearn.com/Course/Python/

Programming—Python - Snakify takes you through from the start in programming if you need a refresher. https://snakify.org/en/lessons/print_input_numbers/

The Alan Turing Institute—"We believe data science and artificial intelligence will change the world.

" https://www.turing.ac.uk

Careers —If you go on to further study Computer Science at Degree level or apprenticeships, here are some links for various opportunities:

https://www.prospects.ac.uk/careers-advice/what-can-i-do-with-my-degree/computer-science

Apprentice opportunities at GCHQ. GCHQ's unique CyberFirst Degree Apprenticeship is a great opportunity to develop advanced cyber security skills, while helping the UK to grow its technical capabilities.

https://recruitmentservices.applicationtrack.com/vx/lang-en-GB/mobile-0/appcentre-3/brand-4/xf-bd8ab30ccf84/candidate/so/pm/1/pl/6/opp/1876

Apprenticeships at MI5—become a technical apprentice "You'll learn some incredible skills. From creating new apps that track terror suspects, to developing innovative ways to stop espionage in its tracks." School Leavers | MI5 - The Security Service