Nonsuch High School for Girls
Sixth Form
September 2026
Computer Science Handbook

“Algorithms will make mistakes. Algorithms will be unfair. That should in no way distract us from the fight to make them more accurate and less biased wherever we can – but perhaps acknowledging that algorithms aren’t perfect, any more than humans are, might just have the effect of diminishing any assumption of their authority.”
— Hannah Fry, Hello World: Being Human in the Age of Algorithms

Welcome to A Level Computer Science

Computer Science at A Level

Computer Science is about designing and using computer-based algorithms to solve problems. This involves computer programming, which forms a significant part of the course, but Computer Science is not just training in a programming language. Instead, the emphasis is on computational thinking. Computational thinking is a kind of reasoning used by both humans and machines: it requires abstraction and decomposition. The study of computation is about what can be computed and how to compute it.

Computer Science involves questions that have the potential to change how we view the world. One can study Computer Science and go on to a career in medicine, law, business, politics or any type of science.

There is a clear distinction between Computer Science and ICT: ICT is about how to use computers, but Computer Science is about how to make computers useful. It has been said that Computer Science is no more about computers than astronomy is about telescopes: this may be an over-statement, but you will find yourself studying aspects of Computer Science theory and mathematics which will feel very remote from one's everyday use of computers.

Computer Science, with its emphasis on abstract thinking, general problem-solving, algorithmic and mathematical reasoning, scientific and engineering-based thinking, is a good foundation for understanding and solving a wide range of future challenges.

In what different ways will you be expected to work?

Students who have found a GCSE Computer Science course straightforward must be prepared for a considerable leap in difficulty as they come to A-level Computer Science. Those who haven't taken Computer Science previously will be expected to do further reading and practice to support their basic understanding. There is much more to learn, in skills, knowledge and techniques than at GCSE but in a way that looks at areas in detail. Students are expected use their study time to go beyond what is taught in class to fully understand and apply the content.

Successful students are those with the focus and determination to succeed no matter what your prior experience is. Problem solving skills are paramount, and these can only be developed through practice solving difficult problems! Students will need to spend time on their own completing tasks out of lessons or undertaking further challenges around similar problems. Successful coders are those able to maintain focus on a tricky problem, or who solve a problem quickly and then go on to practice applying and extending what has been learnt in a different context.

You will be introduced to a large range of advanced algorithms in lessons, but you will certainly not learn and be able to apply them without considerable commitment of your own time. You will have to evidence your journey of learning code independently in the year 13 NEA project. All students will be expected to be able to learn to program in C# as it is industry standard and enables the use of higher-level skills in the project. To support this, you will be given a coding book of ideas for programs.

Sixth Form Computer Science Students Should...

Be Punctual

All students are expected to arrive at lessons on time. There is a lot to get through, and you need to be on time to understand the topics from the ground up. Work will be set on MS Teams for homework, and you are expected to complete it by the deadlines set. Not all work will be formally marked and, instead, discussed and built on in class. This means you need to have attempted the work in order to participate and benefit from the discussion.

Plan Ahead

All students should notify their teacher if they know that they will be absent from a lesson. If the absence is unforeseen, you must contact the teacher before the next lesson and ensure that any missed work is completed.

Be Engaged

All students should come to the lessons fully prepared, with the correct resources, and prepared for effective note-taking. You should have done all the work required in order to engage meaningfully with and benefit fully from the topics covered in the lesson. You are expected to discuss your critical thinking as well as presenting findings to the rest of the class.

Use Their Time Wisely

For every hour you spend in class, you should spend approximately an hour outside of class, at home or during your study periods, consolidating the material you have covered. This is to be managed independently as teachers will expect you to start becoming more self-sufficient.

Have Access To A Computer

You will have to work on programming tasks or other computer-based activities in your own time and will therefore need access to a reasonably up-to-date computer. You will have to install the Visual Studio software, which will be available without charge, but requires either a desktop computer or a laptop computer. Ideally it would be a Windows-based computer to make it more transferable. It will not be possible to complete the Year 13 project sufficiently using a tablet or just within school hours.

Get Involved

In the school year, we ask you to join us at the Sixth form Opportunities evening as you are the ones with the most recent experience and advice to offer those who are considering Computer Science at A-Level. We also ask for volunteers in the summer term with Open Evening.

Go Beyond The Curriculum

Life after A-levels is much better when you already know a bit about the wider world that you are heading towards. You should sign up for journals, e-mails and groups of interest in order to extend and support your learning. There are also many books in the school library of relevance.

Overview of the course

Please be aware that topics and timings during the course are subject to change.
Please follow staff guidance.

	Year 12 Computer Science Theory	Year 12 Programming Theory	Year 12 Programming Practices
Autumn	1.1: Components of a Computer ❖ Input, Output and Storage Devices ❖ Storage and Memory ❖ Processor Components ❖ Processor Performance ❖ Types of Processors	1.4.1: Data Types ❖ Positive and Negative Binary Numbers ❖ Subtraction of binary numbers in 2's complement ❖ Floating Point Binary Numbers ❖ Floating point arithmetic ❖ Bitwise manipulation and masks ❖ Hexadecimal ❖ Character Sets 1.4.2: Abstract Data Types ❖ Primitive Data Types ❖ Arrays, records, lists, tuples ❖ Linked-lists ❖ Queues, stacks ❖ Hash Tables, Dictionaries ❖ Binary Trees	2.1 Elements of Computational Thinking ❖ Thinking Abstractly, Ahead, Procedurally, Logically and Concurrently 2.2: Problem solving and programming ❖ Building on the bridging work through a series of programs. ❖ Use of C# in Visual Studio ❖ Arrays, Records, Lists ❖ Abstract data structures ❖ Functions and procedures ❖ Passing parameters ❖ Global vs local variables ❖ Recursion

Spring	1.2: Software and Software Development ❖ Operating System Functions ❖ Types of Operating Systems ❖ The Nature of Applications	1.4.2: Abstract Data Types ❖ Graphs, Trees ❖ Traversal of graphs ❖ Breadth-first ❖ Depth-first ❖ 2.2: Problem Solving and programming ❖ Algorithms for Abstract Data Types ❖ Adding, deleting, searching ❖ Shortest path Algorithms ❖ Dijkstra's Algorithm ❖ A* Algorithm ❖ 2.3: Algorithms ❖ Efficiency of Algorithms ❖ Big O Notation ❖ Complexity of algorithms ❖ Standard Algorithms ❖ Linear, Binary Search ❖ Bubble, Merge, Quick sort	2.2: Problem solving and programming ❖ Files ❖ Object-oriented Programming ❖ Forms 1.2 - Object-oriented Programming ❖ Classes ❖ Objects ❖ Methods, Attributes ❖ Inheritance, Encapsulation, polymorphism
Summer	Revision and Exams 1.2: Software and Software Development ❖ Software Lifecycles 1.5 Impacts of Technology ❖ Computing Related Legislation	Revision and Exams NEA preparation	Revision and exams Project practice – building an app

Holidays	Building work – wider reading	Component 03 NEA [50 total hours] ❖ NEA Analysis [6 hours]	Building Work

	Year 13 Computer Science Theory	Year 13 NEA	Year 13 Programming
Autumn	1.4.3: Boolean Algebra 1.2: Software and Software Development ❖ Little Man Computer ❖ Programming Language Translators 1.3: Exchanging Data ❖ Databases Theory, Structure, Normalisation 1.5 Impacts of Technology	Component 03 NEA [50 total hours] ❖ Analysis [4 hours] ❖ Design [10 hours] ❖ Development + Testing [15+5 hours] ❖ Evaluation + final tests [8 hours]	2.2 Problem Solving and programming ❖ Computational Methods 1.3: Exchanging Data ❖ Web Technologies ❖ Encryption and Hashing
Spring	1.3: Exchanging Data ❖ Databases – SQL 1.5 Impacts of Technology Revision and Mocks	Component 03 NEA [50 hours] ❖ Evaluation + final tests [2 hours] ❖ Tidy up and submit – first week back 1.3: Exchanging Data ❖ Compression Revision and Mocks	1.3: Exchanging Data ❖ Networks Revision and mocks
Summer	Revision and final exams	Revision and final exams	Revision and final exams

Specification Summary

A Level Computer Science : OCR H446

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

The specification aims to 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
- less emphasis on ICT.

Examination Overview

Paper 1: Computer Systems	
40% of total A-level 140 Marks	2.5 hours Written Paper – no calculators
1.1 Components of a Computer 1.2 Software and software development 1.3 Exchanging Data 1.4 Data Types, Data Structures and Algorithms 1.5 Legal, Moral, Cultural and Ethical Issues	
Paper 2: Algorithms and Programming	
40% of Total A-level 140 marks	2.5 hours Written Paper – no calculators
2.1 Elements of computational thinking 2.2 Problem solving and programming 2.3 Algorithms	
Component 3: NEA Programming Project	
20% of Total A-level 70 marks	Written Report with testing and program code

Assessment Objectives (AO)

AO1	Demonstrate knowledge and understanding of the principles and concepts of computer science, including abstraction, logic, algorithms and data representation
AO2	Apply knowledge and understanding of the principles and concepts of computer science including to analyse problems in computational terms
AO3	Design, program and evaluate computer systems that solve problems, making reasoned judgements about these and presenting conclusions

Component	% of A Level Computer Science (H446)			
	AO1	AO2	AO3	Total
Computer systems	21	9	10	40
Algorithms and programming	15	18	7	40
Programming Project	0	3	17	20
Total	35	30	35	100

Types of Questions

Both exam papers have a range of question styles including stating, describing, explaining and justifying. There are a far number of "Quality of Response" questions that are marked according to how well you cover AO1, AO2 and AO3 for a given context so be prepared to be writing!

Final Project (NEA)

Students are expected to:

- produce a slimmed down programming project which is more refined and more focussed on coding
- choose the project title and problem to be solved
- record their progress, discussing their successes, attempts and failures within the context of their project
- include agile methods.

Student Support

Students will regularly review, reflect and improve on their work and will be encouraged to ask questions and seek clarification on how to improve. If required, we will help you to organise a study group from those in the class during your study periods or lunchtimes. If we have genuine concerns about your progress, you will be asked to meet with the Head of Department to discuss how we can help you to meet your target grades. Bear in mind, that the jump from GCSE to A-Level is a large one, and while it isn't definite, there is a general correlation between GCSE grades 7-9 and A-Level grades D-A*.

What To Do When You Get Stuck

"As soon as we started programming, we found to our surprise that it wasn't as easy to get programs right as we had thought. Debugging had to be discovered. I can remember the exact instant when I realized that a large part of my life from then on was going to be spent in finding mistakes in my own programs."
- Maurice Wilkes discovers debugging, 1949

When you are stuck, your first question should be 'have I spend long enough on this?'. If you have not attempted a solution, you will not learn much from one you are shown. There is a wealth of support material on SharePoint and online. When coding, you will frequently need to look for coding samples on the Internet. www.stackoverflow.com is a great place. Don't post a question until you have searched – it is very unlikely that your problem hasn't been solved and explained already. You should be prepared to support your peers and get help in return, both in class and out. You can seek help from your teachers via MS Teams. Never be afraid to ask a question publicly – if you are stuck, there is a very high chance that others are either in the same place or have worked it out and can help. Don't expect your teachers to know all the answers: by the end of the NEA your project will be as unique as you are!

Resources and wider reading

In-School Resources

We provide an electronic copy of the PG Online Textbook, but as yet haven't found a textbook that fully covers the specification so you should be prepared to use a range of resources to understand the concepts and then learn any specific definitions and answer formats afterwards. The Computer Science department has a variety of textbooks to support learning. These are available in the library. There are also other, more fun, Computer Science related books available to borrow. If you find any other resources that you feel are useful, please let your teacher know. You will be provided with a subscription the revision service – Smart Revise.

You should keep up-to-date with the latest technology and ethical arguments around emerging technologies through magazines and social media platforms. On the Computer Science SharePoint landing page, there is a page called "Going Beyond" which contains a range of suggestions for you to explore. You will be expected to be able to introduce a discussion on emerging technology in lessons.

Conferences and Training

Various conferences, technology shows and corporate opportunities are held throughout the year within London and nearer counties as well as continuing to be available online for those who can't go in person. Your teacher will aim to keep you up-to-date via Teams, however it is often useful to subscribe to online technology newsletters for further details. Sometimes exciting opportunities, competitions and useful workshops are advertised via such newsletters.