Curriculum Intent

In Computing, you learn about how computer systems work from fundamental Binary and Boolean logic to architecture and hardware to the programs and apps which run on them. The course is very practical and skills based and students will also learn how to create algorithms and computer programs and will also significantly develop problem solving skills when doing this. Students learn to program in Visual Basic and will also learn how to query and update databases using SQL.

The vast majority of us now use computers in our day to day lives for everything from gaming, leisure and communication to social media, finding information, paying our bills and shopping. It is a very important field for the future and programmers will be needed in every industry including manufacturing, agriculture medicine, fashion, leisure and retail.

Aims and learning outcomes:

- Build on their knowledge, understanding and skills established through the computer science elements of the programme of study for computing at Key Stage 3
- Enable students to progress into further learning and/or employment
- Understand and apply the fundamental principles and concepts of computer science, including abstraction, decomposition, logic, algorithms, and data representation
- Analyse problems in computational terms through practical experience of solving such problems, including designing, writing and debugging programs
- Think creatively, innovatively, analytically, logically and critically understand the components that make up digital systems, and how they communicate with one another and with other systems
- Understand the impacts of digital technology to the individual and to wider society
- Apply mathematical skills relevant to computer science

Assessment objectives:

AO1: Demonstrate knowledge and understanding of the key concepts and principles of computer science.

AO2: Apply knowledge and understanding of key concepts and principles of computer science. AQA GCSE Computer Science 8520.

AO3: Analyse problems in computational terms: to make reasoned judgements and to design, program, evaluate and refine solutions.

Curriculum Implementation

We cover two strands in Computing, Programming and Theory which as well as developing their knowledge and understanding of Computer Science, will ultimately prepare the students for the paper 1 (Programming) and paper 2 (Theory) exams.

For theory, we aim to cover all theoretical units in year 10 and plan to revisit them in year 11 with a larger focus on fine tuning of subject knowledge and the ability to recall the information. For the Programming strand, we focus on teaching students to program in Visual Basic in year 10 and then move across to paper based programming tasks in year 11.

Summary learning journey for theoretical elements covered in year 10, revisited in year 11:

Topic 1

- Data representation: Number bases
- Data representation: Units of information
- Data representation: Binary Arithmetic & character encoding
- Data representation: Representing images and sound
- Data representation: Data compression

Topic 2

- Fundamentals of networks: Computer networks
- Fundamentals of networks: Topologies
- Fundamentals of networks: Security
- Fundamentals of networks: Protocols and layers

Topic 3

- Hardware and software: Boolean logic
- Hardware and software: Software classification
- Hardware and software: Programming language translators
- Hardware and software: Systems architecture

Topic 4

- Impacts of difital technology: Ethics and environment

Topic 5

- Cyber Security: Fundamentals
- Cyber Security: Threats
- Cyber Security: Social Engineering
- Cyber Security: Malicious code
- Cyber Security: Detection and prevention

Topic 6

- Relational databases & SQL

Year	Term	Content	Rationale
10	Autumn Term 1&2 Theory Strand	**3.3 Fundamentals of data representation** 3.3.1 Number bases 3.3.2 Converting between number bases 3.3.3 Units of information 3.3.4 Binary arithmetic 3.3.5 Character encoding 3.3.6 Representing images 3.3.7 Representing sound 3.3.8 Data compression **Data Representation end of unit test**	Fundamental concept that students are familiar with from KS3 (Prior learning to reactivate) which we then begin with to build confidence. This topic area always comes up in the exam and is very application of skill based (A02), and early coverage highlights any students who need early intervention. We revisit this again in year 11 as the skills require re-enforcement.
10	Spring term 1 Theory Strand	**3.5 Fundamentals of computer networks** 3.5 Computer networks 3.5 Network topologies 3.5 Network security 3.5 Protocols and layers **Networking end of unit test**	A concept which has already been covered in year 8 (Prior learning to reactivate) we build on this knowledge in order to equip students to answer (A01) questions. Again a familiar concept to cover early as this builds confidence. We re-cover these units in year 11 and with a focus on past paper questions and on making revision materials.
10	Spring Term 2 Theory Strand	**Hardware and software** 3.4.2 Boolean logic 3.4.3 Software classification 3.4.4 Classification of programming languages and translators 3.4.5 Systems architecture **Hardware and Software end of unit test**	A number of the concepts within this unit build upon knowledge from KS3 (Prior learning to reactivate) but additional depth is added. Again these are bread and butter A01 and A02 areas of theory which students need to be able to do in the exam. We revisit these areas in year 11 with a focus on Past Papers.
10	Spring Term 2 Theory Strand	**3.8 Impacts of digital technology on wider society** Ethical, legal and environmental impacts **Ethics Question assessment**	Again students have practised writing ethics essay questions in KS3 (Prior learning to reactivate), we rebuild on this knowledge by getting students to explore ethical, legal environmental issues in a number of current ICT scenarios such as the internet of things, driverless cars, google street view, government surveillance and social media. A01 and A02
10	Summer Term 1 Theory Strand	**3.6 Cyber security** 3.6.1 Fundamentals of cyber security 3.6.2 Cyber security threats 3.6.1.1 Social engineering	Covered extensively in KS3 (Prior learning to reactivate) this AO1 area requires students to be able to recall knowledge relating to Cyber Security.

		3.6.1.2 Malicious code (malware) 3.6.3 Methods to detect and prevent cyber security threats **Cyber Security end of unit test**	We revisit all of these areas in this discussion based unit and also look at some current examples in the news. Again this area is revisited in year 11 with past paper questions.
10	Summer Term 2 Theory Strand	**3.7 Relational databases and SQL** 3.7.1 Relational databases 3.7.2 Structured query language (SQL) **Relational databases and SQL end of unit test**	New to this specification, we have covered databases in KS3 (Prior learning to reactivate) and have looked briefly at SQL. This is revisited in more depth from a theoretical point of view here, although this is also covered within practical programming sessions as well. Covered at the end of the year 10 to allow the theory to sink in with students over the Summer and be relatively fresh when it is revisited in year 11.

Programming Strand year 10

Year	Term	Content	Rationale
10	All year via a series of practical projects	**3.2 Programming** 3.2.1 Data types 3.2.2 Programming concepts 3.2.3 Arithmetic operations in a programming language 3.2.4 Relational operations in a programming language 3.2.5 Boolean operations in a programming language 3.2.6 Data structures 3.2.7 Input / output 3.2.8 String handling operations in a programming language 3.2.9 Random number generation in a programming language 3.2.10 Structured programming and subroutines 3.2.11 Robust and secure programming 3.7 Relational databases and SQL 3.7.1 Relational databases 3.7.2 Structured query language (SQL)	Taught across all of year 10 for a minimum of 1 hour per week dependent on class pace. Students learn to program in Visual Basic and will slowly build up all of the content required in the specification. This builds upon programming done in KS3 in Small Basic which has been chosen specifically because it leads into Visual Basic. (Prior learning to reactivate) All students will have the minimum level of programming ability required to access GCSE past paper questions in year 11 but a number of more able students will be given significant stretch and challenge to prepare them for A Level.

Year 11 : Two lessons per week

11	Autumn Terms 1 & 2	**Revisiting essential subject knowledge:** **Theory Strand:** 3.3 Fundamentals of data representation 3.5 Fundamentals of computer networks 3.4 Hardware and software 3.8 Impacts of digital technology on wider society 3.6 Cyber security 3.7 Relational databases and SQL **Programming Strand:** An emphasis on VB written code and how to organise it on paper. Dry running algorithms and using trace tables.	Recap on all topic areas covered in year 10 (Prior learning to reactivate) Focus here will be to revisit this theory but with a greater emphasis on written answers and knowledge recall. Programming will move from the practical work in VB done in year 10 to paper based programming tasks.
11	Spring Term 1	**Mock Exam:** Theory strand: Paper 2 Programming strand: Paper 1	Both Mock Exams are full past papers which will be chosen to cover off all areas of the specification. Results from both exams will be analysed in detail and results of the analysis will inform teaching for the remainder of the year. Teaching will then be tailored to fit areas of the specification which students would benefit from covering again or which we can see require re-enforcement and scaffolding. We will also clarify any common misconceptions. Mock results will also allow us to target students for intervention.
11	Spring Term 2 and Summer 1	Re-teaching of areas which mock exam analysis highlights as requiring re-enforcement. Knowledge recall and revision tasks, timed questions, walking talking mock questions. Intervention sessions for those targeted for intervention.	Our revision program is designed to solidify knowledge and to help students to fine tune their written technique.

Assessment:

Throughout the course, students will complete an end of unit assessment at the end of each half term in order to monitor progress and target intervention.

Students will sit a mock exam at the end of year 10 and then a full mock covering both paper 1 and paper 2 after Christmas in year 11.

Final exam assessment is broken up into two papers.

Paper 1: Computational thinking and programming skills
What's assessed Computational thinking, code tracing, problem-solving, programming concepts including the design of effective algorithms and the designing, writing, testing and refining of code. The content for this assessment will be drawn from subject content 3.1 and 3.2 above.
How it's assessed • Written exam: 2 hours • 90 marks • 50% of GCSE
Questions A mix of multiple choice, short answer and longer answer questions assessing programming, practical problem-solving and computational thinking skills.

Paper 2: Computing concepts
What's assessed The content for this assessment will be drawn from subject content 3.3 to 3.8 above.
How it's assessed • Written exam: 1 hour 45 minutes • 90 marks • 50% of GCSE
Questions A mix of multiple choice, short answer, longer answer and extended response questions assessing SQL programming skills and theoretical knowledge.

Assessment objective weightings for GCSE Computer Science

Assessment objectives (AOs)	Component weightings (approx %)		Overall weighting (approx %)
	Paper 1	Paper 2	
AO1	7	30	35–40
AO2	28	20	45–50
AO3	15	0	15–20
Overall weighting of components	50	50	100

Further Curriculum Support

All topics are covered by the YouTube channel: Craig”n”Dave.

Zig Zag revision guide is available on in student revision zone

Past papers and specification can be found on the AQA GCSE Computer Science Website

and on the revision zone

Visual Studio Community edition can be downloaded from Microsoft.com

Online programming tutorials are available such as Home and Learn visual Basic

Isaac Computer Science website also offers a helpful range of revision materials.

Where can GCSE Computer science take me?

Computing develops skills in computational and critical thinking, analysis, problem solving, initiative and lateral thinking. Computer science complements most subjects which require critical thinking but sits particularly well with maths because it includes significant logical thinking and problem solving.

Computer Science at GCSE helps prepare students for A Level Computer Science and ultimately could lead to careers in Computer programming, software engineering, website and app development, computer game development and cybersecurity. Computer science can also help with other career paths requiring ICT or critical thinking and problem solving skills.