Curriculum intent

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. Computer Science is a very important field for the future and programmers will be needed in every industry including manufacturing, agriculture medicine, fashion, leisure and retail. Relevant to the classroom and the real world, advances in computing are transforming the way we work. The course is very practical and skills based and as well as learning a broad range of Computer Science theory, students will also learn how to create algorithms and computer programs and will also significantly develop problem solving skills when doing this. Students develop their skills in Visual Basic from GCSE and will ultimately be able to implement a complex programming project and be able to interact with complex exam board provided skeleton code.

Course content:

- Fundamentals of programming
- Fundamentals of data structures
- Fundamentals of algorithms
- Theory of computation
- Fundamentals of data representation
- Fundamentals of computer systems
- Fundamentals of computer organisation and architecture
- Consequences of uses of computing
- Fundamentals of communication and networking
- Fundamentals of databases
- Big Data
- Fundamentals of functional programming Systematic approach to problem solving
- Non-exam assessment - the computing practical project

Course implementation

We have split A level computing into two stands: Theory and Programming.

In order to allow students to access shared expertise across the department, the A level content is usually shared between either two or three teachers and we have planned delivery in line with this. We have found this approach to be very effective as staff are specialists in different areas, and students benefit from having a point of contact with a number of teachers particularly when needing to access advice on the programming projects.

Paper 2 Theory:

We run two theory strands at once and additionally we run a practical programming strand. We have found that this is a useful way to cover ground effectively but we do make a careful choice of units so that students are able to cope with this.

We cover a large amount of the theory in year 12 and much of year 13 is top up with a limited amount of extra content with a bigger focus on past papers and mark schemes.

Programming and Paper 1 Programming Theory:

Much of year 12 is spent on teaching students to program beyond GCSE level in Visual Basic console. During this time, the theoretical programming concepts are also delivered practically and students

are exposed to Skeleton code. At the end of year 12 teaching moves to forms and databases so that students are equipped to work on the NEA in year 13. Programming theory is revisited in year 13 from a more theoretical standpoint and time is also spent familiarising students with the pre-release skeleton code.

Year 12

Paper 2 Theory: 2 hours per week

Year	Hours per week	Term	Content	Rationale
12	1	Autumn Term 1&2 Teacher 1	**4.9 Fundamentals of communication and networking** 4.9.1 Communication 4.9.2 Networking 4.9.3 The Internet 4.9.4 (TCP/IP) protocol **Networking end of unit test**	Straight forward theory unit which builds upon knowledge from GCSE and KS3 Computer Science. Good confidence builder as a starting point as it is largely knowledge based – good running alongside the other theory unit and programming unit.
12	1	Autumn Term 1&2 Teacher 2	**4.7 Fundamentals of computer organisation and architecture** 4.7.1 Internal hardware components of a computer 4.7.2 The stored program concept 4.7.3 Structure and role of the processor and its components 4.7.4 External hardware devices **Architecture end of unit test**	Builds upon knowledge from GCSE Computer Science and KS3 Computer Science but there is more complexity in this unit, particularly surrounding the processor registers, FDE Cycle and Assembly Language
12	1	Autumn Term 2 Teacher 1 and Teacher 2	**4.8 Consequences of uses of computing** 4.8.1 Individual (moral), social (ethical), legal and cultural issues and opportunities **Ethics question**	Each teacher will do at least one of: discussion and coverage of an up to date computer ethics case study followed by an essay question on that issue. Advice on how to structure an ethics question will be covered. Students are generally very good at these questions and this builds on GCSE and KS3 but it is useful to periodically revisit this area.
12	1	Spring Term 1& 2	**4.5 Fundamentals of data representation** 4.5.1 Number systems	Builds upon known concepts from GCSE Computer Science and KS3, this area is very

		Teacher 1	4.5.2 Number bases 4.5.3 Units of information 4.5.4 Binary number system 4.5.5 Information coding systems 4.5.6 Representing images, sound and other data **Number systems end of unit test**	familiar and it is largely a re-activation and then top up of prior learning. Another confidence builder running alongside a more complex unit encompassing Boolean Algebra
12	1	Spring Term 1& 2 Teacher 1	**4.6 Fundamentals of computer systems** 4.6.1 Hardware and software 4.6.2 Classification of programming languages 4.6.3 Types of program translator 4.6.4 Logic gates 4.6.5 Boolean algebra **Computer systems end of unit test**	Builds upon knowledge from GCSE Computer Science but there is more complexity in this unit, particularly surrounding Boolean algebra. Lots of repetitive tasks required to get students to be able to do this properly.
12	2	Summer Term 1 Teacher 1 Teacher 2	Revision of theory and past paper coverage of AS papers in readiness for the year 12 Mock Exam. Also targeted coverage of any theory in the mock that falls outside of units taught. **Mock Exam**	It is important to prepare students for their mock exams and to make sure that they are familiar with the structure of exam papers and the way in which they are marked in readiness for the exams.
12	1	Summer Term 2 Teacher 1	Targeted recap on areas students underperformed in in the mock exam.	Following detailed analysis of the mock exam, areas where students underperformed will be highlighted and these will be retaught in the Summer term in order to support students. Typical areas requiring re-covering are the more complex topics such as Boolean algebra and assembly language.
12	1	Summer Term 2 Teacher 1	**4.10 Fundamentals of databases** 4.10.1 Conceptual data models and entity relationship modelling	In the summer term, in the programming strand, students are working on a developing database driven projects. In line with this, database theory

			4.10.2 Relational databases 4.10.3 Database design and normalisation techniques 4.10.4 Structured Query Language (SQL)	will be looked at to complement the work they are doing within the programming strand. Databases have been previously covered in GCSE Computer Science and BTEC National in ICT and also at KS3 This will be revisited in more detail in year 13

Programming and Programming theory year 12: 3 hours per week

Year	Hours per week	Term	Content	Rationale
12	2	Autumn term 1 & 2 Spring term 1 & 2	**General programming** 4.1 Fundamentals of programming 4.1.1 Programming 4.1.2 Programming paradigms 4.2 Fundamentals of data structures 4.2.1 Data structures and abstract data types 4.2.2 Queues 4.2.3 Stacks 4.2.4 Graphs 4.2.5 Trees 4.3 Fundamentals of algorithms 4.3.1 Graph-traversal 4.3.2 Tree-traversal 4.3.4 Searching algorithms 4.3.5 Sorting algorithms 4.3.6 Optimisation algorithms	Development of programming skills in Visual Basic Console from GCSE and Small basic and Scratch as KS3. Starting with confidence building known programming tasks and building up to more complex programming, data structures will also be considered throughout. Some external students have a background in Python, so need initial support in using VB. Whilst programming is being taught, practical coverage of a number of algorithms such as graphs, trees, searching and sorting algorithms will be included to expose students to these areas but these will be covered more formally from a theoretical point of view in year 13.
12	1	Autumn term 1 & 2 Spring term 1 & 2	**Preparation for paper 1**	Students will have exposure to Paper 1 practical questions

			4.4 Theory of computation **Abstraction and automation** 4.4.2 Regular languages 4.4.3 Context-free languages 4.4.4 Classification of algorithms 4.4.5 A model of computation Working with previous examples of Skeleton code.	such as finite state machines and dry running. Dry running and Pseudocode have been previously visited in KS4 Computer Science. Students will be also be given exposure to a number of AS Skeleton codes so that they are familiar with these in preparation for their programming exam in year 13. Dependant on pace of the group they will move on to work with year 13 Skeleton codes.
12	2	Summer term 1	Preparation for programming mock exam. Timed programming exercises and looking at AS paper 1's in order to allow students to become familiar with these. **Mock Exam**	
12	1	Summer term 1	Visual basic forms, databases and SQL. Preparation for programming project.	Students are shown how to build a menu driven forms application with a database back end in order to give them a starting point on their year 13 project. Menu driven systems and interface design have been visited at KS3 and in BTEC ICT.
12	3	Summer term 2	Visual basic forms, databases and SQL. Preparation for programming project.	

Theory Strand: year 13 typically 2 hours per week

Year	Term	Content	Rationale
13	Autumn Term 1 Teacher 1	**4.10 Fundamentals of databases** 4.10.1 Conceptual data models and entity relationship modelling 4.10.2 Relational databases 4.10.3 Database design and normalisation techniques 4.10.4 Structured Query Language (SQL) **Databases end of unit test**	Touched upon at the end of year 12, this unit will be revisited with a focus on answering exam based questions. This topic always comes up in paper 2, so focussing on how to answer these questions in an exam is key.
13	Autumn Term 1 Teacher 2	**4.5 Fundamentals of data representation – Focus on the A Level element such as floating point and answering exam questions** 4.5.1 Number systems 4.5.2 Number bases 4.5.3 Units of information 4.5.4 Binary number system 4.5.5 Information coding systems 4.5.6 Representing images, sound and other data **Floating Point binary end of unit test**	Covered in year 12 the focus here will be to look at the additional content for A level on top of the AS theory. Particular focus on floating point binary and normalisation as this is a question that always comes up in the exam.
13	Autumn Term 1 Teacher 1	**4.6 Fundamentals of computer systems** - focus on extra elements for A Level such as D type Flip Flops and on answering exam questions. 4.6.1 Hardware and software 4.6.2 Classification of programming languages 4.6.3 Types of program translator 4.6.4 Logic gates 4.6.5 Boolean algebra **Computer systems end of unit test**	Covered in year 12 the focus here will be to look at the additional content for A level on top of the AS theory. Particular focus on past paper questions and extra elements such as D type flip flops.
13	Autumn Term 2 Teacher 2	**4.9 Fundamentals of communication and networking** – recap from year 12 with a focus on past	Covered in year 12 the focus here will be to look at the additional content for A level on top of the AS theory. Particular focus on past paper questions.

		paper questions and additional A level content. 4.9.1 Communication 4.9.2 Networking 4.9.3 The Internet 4.9.4 (TCP/IP) protocol **Networking end of unit test**	
13	Spring term 1 Teacher 1 Teacher 2	Revision of theory and past paper coverage of A2 papers in readiness for the year 13 Mock Exam. Also targeted coverage of any theory in the mock that falls outside of units taught. **Mock Exam**	It is important to prepare students for their mock exams and to make sure that they are familiar with the structure of exam papers and the way in which they are marked in readiness for the exams.
13	Spring term 2 & Summer term 1 Teacher 1 Teacher 2	Targeted recap on areas students underperformed in in the mock exam. Work on past papers and mark schemes and fine tuning theory. Final preparation for paper 2 exam. **Some time may be allocated to students for completion of coursework.**	Following detailed analysis of the mock exam, areas where students underperformed will be highlighted and these will be retaught in the Summer term in order to support students.

Programming strand year 13 (typically 3 hours per week)

Year	Hours per week	Term	Content	Rationale
13	0.5	Autumn term 1 & 2	**4.2 Fundamentals of data structures** 4.2.1 Data structures and abstract data types 4.2.2 Queues 4.2.3 Stacks 4.2.4 Graphs 4.2.5 Trees 4.2.6 Hash tables 4.2.7 Dictionaries 4.2.8 Vectors	Visiting the programming content again from year 12 but this time with a focus on applying these to paper 1 questions on exam papers in readiness for the final exam.

			4.3 Fundamentals of algorithms 4.3.1 Graph-traversal 4.3.2 Tree-traversal 4.3.4 Searching algorithms 4.3.5 Sorting algorithms 4.3.6 Optimisation algorithms **Focus on answering exam questions relating to these areas.**	
	0.5	Autumn term 1&2	**Skeleton Code**	Preparatory work with pre-release skeleton code which is released in September of year 13.
	2	Autumn term 1&2	**Practical Programming Project**	This project although only worth 20% of the final course takes students a large amount of time to complete. This is seen as a beneficial trade off as it prepares students for the practical programming exam. 2 hours per week as a workshop to support students with their NEA. **It is accepted that the year 13 homework is largely work on completion of NEA.**
12	0.5	Spring Term 1&2	**4.11 Big Data** **4.12 Fundamentals of functional programming** 4.12.1 Functional programming paradigm 4.12.2 Writing functional programs 4.12.3 Lists in functional programming Recap on **4.4 Theory of computation** **Abstraction and automation** MOCK EXAM: **Paper 1** **Spring term 1**	This theory has not been covered in year 12 and is often on paper 2 and paper 1 papers, with some language which can catch students out. Focus will be on example past paaper questions. Big data has been discussed at KS4 and KS3.

	0.5	Spring term 1&2	**Skeleton Code**	Preparatory work with pre-release skeleton code
	2	Autumn term 1&2	**Practical Programming Project**	This project although only worth 20% of the final course takes students a large amount of time to complete. This is seen as a beneficial trade off as it prepares students for the practical programming exam. 2 hours per week as a workshop to support students with their NEA. **It is accepted that the year 13 homework is largely work on completion of NEA.** **First completion: Easter**
12	2	Summer term 1	Finishing off of NEA	Students will be given 2 hours per week until their final deadline to complete lose ends of their NEA
12	1 moving to 3 when NEA is complete	Summer term 1	Preparation for practical programming exam.	Important to get students fully ready for the practical programming exam with a focus on making sure they can interact with and answer questions relating to Skeleton code. Final revision of the theoretical elements of the programming exam.

Assessment

Paper 1
What's assessed: this paper tests a student's ability to program, as well as their theoretical knowledge of Computer Science from subject content 10-13 above and the skills required from section 22 above.
Assessed • On-screen exam: 2 hours 30 minutes • 40% of A-level
Questions Students answer a series of short questions and write/adapt/extend programs in an Electronic Answer Document provided by us. We will issue Preliminary Material, a Skeleton Program (available in each of the Programming Languages) and, where appropriate, test data, for use in the exam.

Paper 2
What's assessed: this paper tests a student's ability to answer questions from subject content 14-21 above.
Assessed • Written exam: 2 hours 30 minutes • 40% of A-level
Questions Compulsory short-answer and extended-answer questions.

Non-exam assessment
What's assessed: the non-exam assessment assesses student's ability to use the knowledge and skills gained through the course to solve or investigate a practical problem. Students will be expected to follow a systematic approach to problem solving, as shown in section 22 above.
Assessed • 75 marks • 20% of A-level

Further Curriculum Support

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

Zig Zag revision guide is available on Student Shared area.

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

Visual Studio Community edition can be downloaded from Microsoft.com

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

Course textbook: AQA A level Computer Science: Bob Reeves

Where does the course lead to?

The ability to program is a very marketable skill, especially looking forwards.

Computer science can lead on to a number of different degree paths such as robotics, artificial intelligence, machine learning, cloud computing, big data processing, networking, ethical hacking, computer game development, home automation or even teaching.