Computing Curriculum Overview

Intent

Summary of curriculum intent

In computing, the intent of the curriculum is to ensure all students are equipped with knowledge and skills which allow them to be confident in their use of technology, but also to develop their ability to solve problems and computational thinking through learning to code. The curriculum offer is designed to ensure that practical ICT skills which can be applied in a wide range of contexts are taught early to ensure that students can apply these skills to their other subjects. This is accompanied by programming and computing theory to ensure that students are well prepared to study a range of ICT/Computer Science courses, and Online Safety to support students to make informed and sensible decisions about their use of technology.
Broadly the curriculum can be divided into three areas; Computer Science, ICT and Digital Literacy.

Subject content

Computer Science	Information Technology	Digital Literacy
Algorithms Programming Number Systems Binary Logic Hardware and software Databases Web technologies Data representation Computer Networks Legal, ethical and moral considerations	Use of Office applications Designing web pages Image editing Animation Presenting Data Databases Spreadsheets Design cycle Legal, ethical and moral considerations	Typing Use of email Use websites Advantages & disadvantages of using technology

Computational Thinking Skills

All subjects develop students' ability to apply the following computational thinking skills:

- Decomposition
- Abstraction
- Pattern recognition
- Algorithms

These skills are applied to topics across Computing curriculum; not exclusively in the Computer Science stream.

Implementation

Content and skills in computing is built with the fundamentals of both IT and computer science curriculum learnt first. Concepts are then revisited and built on with greater detail, and increasing complexity as the year progresses. At KS3, the units are varied between Computer Science units and IT units so that students are exposed to both before they decide whether to take it as an option subject in year 9. Lessons work progressively through objectives multiple checkpoints for teachers and students to reflect on their knowledge and skills gained and allowing teachers to adapt as necessary. The curriculum is carefully planned out to ensure that the units covered are relevant to the KS4 and KS5 curriculum and that they are built on from the previous year to consolidate learning.

Literacy is developed through explicitly teaching of keywords, keyword glossaries at the back of KS3 workbooks and KS4 exercise books as well as exam command word and toolkit cards for extended writing (particularly at KS4).

In order to ensure that students of different abilities are able to access the curriculum, a distinction in options are made in year 9 where a select set of students (LAP's) are taught a more IT focussed curriculum to prepare them for the IT qualification at KS4, whilst the HAP's continue to build on the computational thinking and computer science stream. At KS4, we provide the two main routes; GCSE Computer Science and the OCR Information Technologies qualification. In order to meet the needs of all students, a core curriculum of the ASDAN course/WP qualification is provided for those who cannot access the Level 2 content of either the Computer Science or IT qualifications.

Impact

Students are regularly assessed through assessed homework for each unit, as well as mini-starter quizzes at the start of KS4 lessons which are based on any topic covered so far and not necessarily the topic we are currently on – interleaving, in order to build confidence in applying skills in unfamiliar contexts and ensure students are continuously recalling information from previous units. Additionally, students undertake formal exams twice per year – at our quarter 2 and quarter 4 reporting points in December and July. In addition, low stakes assessments are conducted at quarter 1 and quarter 3 reporting points to gauge progress and inform planning/teaching.

The assessments are carefully planned to ensure both content and skills based questions are tested (AO1, AO2 and AO3 skills) which help to ensure that students are not only able to demonstrate their understanding but also able to apply key concepts to a range of contexts.
The information collected from the quarterly assessments are used to create question level analysis documents whereby the topics students more struggled with are easily identifiable and it can be used to adapt revision and re-teach concepts where required. The question level analysis also enables the teacher to provide the student with a tailored list of topics they found challenging and direct them to resources that would help to improve their understanding.

Computer Science Curriculum Intent

Computing/Computer Science Course Overview

Course overview	Year 7	Year 8	Year 9 Computer Science	Year 10 Computer Science	Year 11 Computer Science		Year 12 Computer Science	Year 13 Computer Science
Autumn 1	**Unit 1: Digital Literacy** **Knowledge:** Understanding how to use basic office packages & key functions of a computer **Skills:** Using emailing appropriately, formulas and functions in spreadsheets, formatting word documents	**Unit 1: Web technologies** **Skills:** Using HTML and CSS code to build skills on different components of creating a website – resulting in the students creating their own school website	**Unit 1: Basic python programming** **Skills:** Using text based programming to build basic python skills such as variables, iteration, selection, lists and functions. Builds upon the python Micro:bit unit in year 8	**Unit 1: Networks & Python programming** **Knowledge:** Understanding how computers are connected in a network and how they communicate across the network through the TCP/IP protocol model **Skills:** Using text based programming to build basic python skills such as variables, iteration, selection, lists and functions through small and larger programming problems to solve. Builds upon the basic programming learnt in year 9	**Unit 1: Databases** **Knowledge:** Understanding of what a database is and basic concepts such as a table, record, field and primary key **Skills:** Use SQL to insert and retrieve data from a database		**4.4 Theory of Computation** **Knowledge:** Understand techniques to support problem solving and model solutions. Understand how to classify algorithms by their complexity. **Skills:** Apply problem solving and modelling techniques. Apply regular and context-free languages. Apply maths for regular expressions. Analyse the complexity of algorithms. **4.1.1 Programming** **Knowledge:** Understand the syntax of the Java programming language, and, if continuing from GCSE, builds on knowledge of programming in Python (or other language studied). **Skills:** Apply core Java programming skills to solve basic problems.	**4.9 Fundamentals of communication and networking** **Knowledge:** Understand how computers communicate with peripheral devices and with other computers. Understand the infrastructure of networks and the internet. Understand security threats online. **Skills:** Demonstrate and apply your knowledge of communication and networking. **NEA (see below)**
Autumn 2	**Unit 2: Internet Safety** **Knowledge:** Understanding how to be safe online and the consequences of not doing so. Being able to identify what "fake news" is online	**Unit 2: Computing theory 2** **Knowledge:** Understanding the different components of a computer and how they help to function a computer **Skills:** Ability to calculate binary/decimal and demonstrate 2 level binary addition as well as mathematical calculations of how image and sound file sizes are calculated	**Unit 2: Mathematic for computing** **Knowledge:** Understanding how computers use binary to communicate and how logic gates are used. **Skills:** Ability to calculate binary/decimal and demonstrate 3 level binary addition as well as mathematical calculations of how image and sound file sizes are calculated. The ability to formulate truth tables based on logic gates	**Unit 2: Ethical, legal and environmental issues** **Knowledge:** Understanding the current ethical, legal and environmental impacts and risks of digital technology on society **Skills:** Applying the knowledge of these ethical, legal and environmental issues to a 10-12 mark essay question on any given topic	**Revision**		**4.8 Consequences of uses of computing** **Knowledge:** Understanding the ethical, cultural, legal and environmental issues surrounding the development of digital technology. **Skills:** Applying the knowledge of these ethical, cultural, legal and environmental issues to a 9-12 mark essay question. **4.1.2 Programming Paradigms** **Knowledge:** Understand the principles and advantages of procedural- and object-oriented programming paradigms **Skills:** Able to apply the structured approach to creating programs. Able to write object-oriented programs. Able to draw and interpret class diagrams.	**4.12 Fundamentals of functional programming** **Knowledge:** Understand the main features of the functional programming paradigm. **Skills:** Be able to write programs using a functional language. **NEA** **Skills:** Independently develop practical skills in the context of solving a realistic problem or carrying out an investigation. **Knowledge:** Deepen understanding of Computer Science through practical experience.
Spring 1	**Unit 3: Computing theory 1** **Knowledge:** Understanding how computers communicate including the hardware and software of a computer **Skills:** Ability to calculate binary to decimal and demonstrate how protocols are used to communicate across a network	**Unit 3: Digital images** **Skills:** Ability to manipulate images using software to enhance or refine an image for a particular need.	**Unit 3: Intermediate python programming** **Skills:** Using text based programming to build on the basic python skills from term 1 and using more intermediate skills such as file handling and functions when programming.	**Unit 3: Data Representation & Intermediate python programming** **Knowledge:** Understanding how computers use binary to communicate and understanding how data can be compressed through RLE and Huffman coding **Skills:** Ability to calculate binary/decimal/hexadecimal and demonstrate 3 level binary addition as well as mathematical calculations of how image and sound file sizes are calculated. The ability to demonstrate how files can be compressed through RLE or Huffman coding	**Revision**		**4.10 Fundamentals of Databases** **Knowledge:** Understanding how persistent data is structured, stored and retrieved in relational databases. **Skills:** Able to design relational databases using ER diagrams and normalisation techniques. Able to use Data Definition Languages and Structured Query Languages. **4.5 Data Representation** **Knowledge:** Understanding how computers use binary to store and communicate different types of information. **Skills:** Application of number system conversions and mathematical operations. Calculating storage sizes of image and sound files. Calculating precision and accuracy of data stored.	**Revision** **NEA (see above)**
Spring 2	**Unit 4: Programming with the Micro:bit (Block)** **Skills:** Using variables, iteration and selection with block programming to program the Micro:bit to complete a specific task	**Unit 4: Data Handling** **Skills:** Building on digital literacy skills in year 7 so that students are competently able to use the functions in a spreadsheet and database to handle data	**Unit 4: Networks** **Knowledge:** Understanding how computers are connected in a network and how they communicate across the network.	**Unit 4: Fundamentals of algorithms & Advanced programming** **Skills:** Applying different data search methods and data sorting methods to find specific data as well as being able to applying abstraction and decomposition to complete trace tables based on unknown pseudocode Understanding the various classification of programming languages **Skills:** Using text based programming to apply advanced programming skills to a problem, including robust and secure programming techniques, and structured approaches to programming	**Revision**		**4.6 Fundamentals of computer systems** **Knowledge:** Understand the main element s of relationship between hardware and software that make up a computer system and how programming languages are classified. **Skills:** Apply Boolean algebra and logic gates. **4.2 Fundamentals of Data Structures** **Knowledge:** Understand how to store large volumes of data in formats that make it easy for programs and users to access. **Skills:** Able to implement data structures in Java.	**Revision**

Summer term	**Unit 5: Python turtle programming** **Skills:** Build on the skills from spring 2 to text based programming – using similar skills as learnt in previous unit.	**Unit 5: Programming with the Micro:bit (text based)** **Skills:** Building on skills from year 7 – now using text based programming to use variables, iteration and selection with block programming to program the Micro:bit to complete a specific task	**Unit 5: Cyber Security** **Knowledge:** Understanding the cyber security threats online and precautions that can be taken to prevent these threats	**Unit 5: Cyber Security** **Knowledge:** Understanding the cyber security threats online and precautions that can be taken to prevent these threats **Unit 6: Hardware and Software** **Knowledge:** Understanding the functions of an operating system and be able to explain the Von Neumann architecture to demonstrate how the fetch decode execute cycle work to get data from the main memory	**Revision**		**4.7 Fundamentals of computer organisation and architecture** **Knowledge:** Understand the role of internal and external hardware components and the operation of the processor. **Skills:** Able to read and write simple assembly language code. **4.3 Fundamentals of Algorithms** **Knowledge:** Understand how to traverse data structures introduced in 4.2 and the complexities of each algorithm. Understand Reverse Polish Notation. **Skills:** Be able to trace and analyse the complexities of given algorithms and evaluate RPN expressions. **NEA (see next column)**	**Revision**

Computer Science Curriculum Impact

	Q1	Q2	Q3	Q4
Year 7		Digital Literacy Online Safety	Digital Literacy Online Safety Computing theory Programming with micro:bits	Digital Literacy Online Safety Computing theory Programming with micro:bits Python turtle
Assessed homework	Digital literacy	Online Safety	Computing theory 1 Programming with microbits	Python turtle
Year 8		Web technologies Computing theory 2	Web technologies Computing theory 2 Graphics Data handling	Web technologies Computing theory 2 Graphics Data handling Python programming with micro:bit
Assessed homework	Web technologies	Computing theory 2	Graphics Data Handling	Python programming with micro:bits
Year 9 CS		Data Representation	Data Representation Basic Programming Intermediate programming Networks	Basic programming Intermediate programming Mathematics for computing Networks Cyber Security
Assessed homework	Web technologies	Computing theory 2	Graphics Data Handling	Python programming with micro:bits
Year 10 CS	• Networks • Basic programming	**Paper 1:** Programming **Paper 2:** Networks Ethical, legal and environmental	• Algorithms • Programming • Data Representation • Cyber security • Ethical, legal and environmental • Networks	**Paper 1:** Programming Algorithms **Paper 2:** Networks Ethical, legal and environmental Data Representation Cyber security Hardware and software
Assessed homework	Networks	Ethical, legal and environmental	Data Rep Algorithms Cyber security	Hardware and software
Year 10 WJEC	Milestone marker – coursework review grade	Milestone marker – coursework review grade	Milestone marker – coursework review grade	First unit
Assessed homework	LO1	LO1	N/A	LO4
Year 11 CS	All content – 1 paper	All content – 2 papers	All content – 1 paper	N/A
Year 11 IT	LO1,LO3, LO4	All content	N/A	N/A
Year 12	• Programming • Fundamentals of computer systems	**Paper 1:** Fundamentals of Programming Programming Paradigms **Paper 2:** Fundamentals of computer systems Consequences of uses of computers	**Paper 1:** Fundamentals of Programming Programming Paradigms Theory of computation Data Structures **Paper 2:** Fundamentals of computer systems Consequences of uses of computers Fundamentals of databases Data Representation	**Paper 1** Fundamentals of Programming Programming Paradigms Theory of computation Data Structures Fundamentals of algorithms **Paper 2** Fundamentals of computer systems Consequences of uses of computers Fundamentals of databases

				Data Representation Computer organisation
Assessed homework	Fundamentals of computer systems Fundamentals of programming	Theory of computation Programming paradigms	Data Representation Consequences of computing Fundamentals of computing	Fundamentals of algorithms Computer systems Data structures
Year 13	**Paper 1** All content **Paper 2** All content	**Paper 1** All content – including prelim material **Paper 2** All content	**Paper 1** All content – including prelim material **Paper 2** All content	N/A