Computer Science Curriculum Map

Key knowledge & skills to be mastered by students				Year 7		
	Learning Period 1:Autumn	**Learning Period 2:Autumn**	**Learning Period 3: Spring**	**Learning Period 4: Spring**	**Learning Period 5: Summer**	**Learning Period 6: Summer**
Topic title	Using Technology Safely	Data Modelling (Spreadsheets	Computer Networks	Scratch Programming 1	Multimedia 1	Computer Systems
Core principle	Use a wide range of software and technology safely Embed fundamental ICT skills of word processing Create and edit a variety of media Be aware of the risks of technology and how they can be minimised Maximising the use of Horizons	Use a wide range of software and technology safely Maximising the use of Horizons	Understand what networks are and how they are used Recognise and predict technology trends Maximising the use of Horizons	Use a wide range of software and technology safely Modify and create computer programs Use computational thinking skills to solve real world problems Understand simple Boolean logic Maximising the use of Horizons	Use a wide range of software and technology safely Embed fundamental ICT skills Create and Edit a variety of media Be aware of the risks of technology and how they can be minimised Recognise and predict technology trends Maximising the use of Horizons	Use a wide range of software and technology safely Embed fundamental ICT skills Maximising the use of Horizons
Declarative Knowledge (Facts, rules, principles, and the relationships between them)	Knowledge of the definition of HTTPS Knowledge of how to use social media ready Identify different programs for different tasks Know what constitutes a secure password Private information is about an identifiable person Public information is information that can be shared with everyone. Knowledge of the structure of a good presentation Know the differences and similarities between working on iPads and Computers	Knowledge of cell references Knowledge of formulas (all start with equals) Knowledge of the types of charts available Know what a spreadsheet is Know the uses of spreadsheets Knowledge of SUM, MAX, COUNTIF, MIN, AVERAGE, COUNTA Knowledge of discrete and continuous data Knowledge of data collection methods Understand primary and secondary data types	Know what is a network Difference between LAN and WAN Hardware used in a LAN What is the Internet, how is it connected Advantages and disadvantages of wired and wireless Factors that affect performance of networks Know what the Internet of Things is Know what services are available online Knowledge of what a packet is Knowledge of how networks use addresses	Knowledge of Sequence Knowledge of Inputs and Outputs Knowledge of Selection structure (If elif else) Knowledge of Variables Knowledge of forever loops and iteration	Knowledge of plagiarism Knowledge of copyright law Knowledge of information sources and fake news Knowledge of formatting tools and programs Knowledge of what email is	Knowledge of the names and roles of computer components Know the difference between hardware and software Understand that 1 archetype of software has lots of products. (MS word is 1 example of a word Processor) Knowledge of peripherals Knowledge of types and units of storage Know what an operating system is
Procedural Knowledge (Knowledge of methods and processes that can be performed)	Use input devices effectively Use word processors effectively Use presentation processors effectively Use web browsers effectively Use email effectively Determine which information in an extract is personal Use iPads with care and respect effectively Ability to deliver a presentation from notes and PowerPoint	Able to lay data out in a spreadsheet so that it can be analyzed Able to use spreadsheet software on iPad and PC Able to format spreadsheets with borders and conditional colouring Able to change data type in a spreadsheet Able to use auto-fill effectively Able to collect data using a survey	Able to draw a network from a description Able to draw a network from a requirements Able to recognise an IP or MAC addresses Able to	Able to use scratch to make simple programs Able to break down a problem into smaller tasks and plan a program Able to track the changing value of a variable through a program Able to trace the control flow through a simple program (run the program in their heads)	Able to use PowerPoint, word, and photopea to make text and image media. Able to evaluate the credibility of a source Able to write an email with subject and recipient	Be able to label a diagram of a computer Be able to recognise a part from a picture, common features, port type. Able to sort peripherals into inputs and outputs Able to list some features/purposes of an operating system
Assessment & Educational Visit Opportunities	End of Topic formative	End of Topic Formative	End of Topic Formative	End of Topic Formative	End of Topic Formative	End of Topic Formative
Lesson Sequence	1. Passwords and logging on 2. Getting started with Horizons 3. Respectful Communication Online 4. What is cyberbullying? 5. Presenting to an Audience 1 6. Presenting to an Audience 2	1. What is a spreadsheet? 2. Quick Calculations 3. Collecting Data 4. Working with Data 5. Working with Data 2 6. Bringing it all together	1. Introduction to Networks 2. Network Hardware 3. Wired and Wireless Networks 4. The Internet 5. The internet of Things 6. The World Wide Web Safer Internet Day (Optional)	1. Sequence and Variables 2. Selection 3. Operators 4. Iteration 5. Problem Solving 6. Bringing it all together	1. Word Processing 2. Licencing and copyright 3. Credibility 4. Email and online images 5. Putting content online 6. Assessment	1. Peripherals 2. Internal Hardware 3. Storage 4. Application software 5. System software 6. Bringing it all together

Computer Science Curriculum Map

Key knowledge & skills to be mastered by students				Year 8		
	Learning Period 1:Autumn	**Learning Period 2:Autumn**	**Learning Period 3: Spring**	**Learning Period 4: Spring**	**Learning Period 5: Summer**	**Learning Period 6: Summer**
Topic title	Scratch 2	Binary logic and Booleans	Computational Thinking and Algorithms	Programming with Python	Multimedia 2	Developing for the web
Core principle	Use a wide range of software and technology Modify and create computer programs Use computational thinking skills to solve real world problems Understand simple Boolean logic Maximising the use of Horizons	Understand simple Boolean logic Maximising the use of Horizons	Use a wide range of software and technology Embed fundamental ICT skills Use computational thinking skills to solve real world problems Understand simple Boolean logic Maximising the use of Horizons	Use a wide range of software and technology Modify and create computer programs Use computational thinking skills to solve real world problems Understand simple Boolean logic Maximising the use of Horizons	Use a wide range of software and technology Create and edit a variety of media Maximising the use of Horizons	Use a wide range of software and technology Create and edit a variety of media Be aware of the risks of technology and how they can be minimised Understand what networks are and how they are used Recognise and predict technology trends Maximising the use of Horizons
Declarative Knowledge (Facts, rules, principles, and the relationships between them)	Know how to construct conditions for selection Know how to use count and condition controlled loops Know how to make programs with multiple sprites Know how to make a game Know the difference between functions and procedures Know how parameters are passed to subroutines	Knowledge of AND, OR, and NOT Know difference between operands and operators Know 1 and 0 are bits Know the purpose of logic gates Know the layout of truth tables	Knowledge of abstraction, algorithmic thinking, decomposition, pattern recognition Knowledge of what each symbol in a flowchart is for Knowledge of pseudocode structures Knowledge of Linear and binary search Knowledge of basic sorting algorithms Knowledge of uses for sorting algorithms	Knowledge of syntax requirements Knowledge of types of error Knowledge of basic data types Knowledge of Selection Structures Knowledge of while loops Knowledge of variables in a python	Know the difference between bitmaps and vectors Know how frames and images relate to videos and animations Know how colour depth and resolution affect bitmaps Know some of the uses and limitations of vectors	Knowledge of how websites are constructed using HTML and CSS Know how websites are accesed using hyperlinks and webservers Know how websites include images and videos Know why it is important that everyone has access to websites.
Procedural Knowledge (Knowledge of methods and processes that can be performed)	Be able to make a simple real time game with keyboard controls Be able to make code simpler and more maintainable with subroutines	Be able to construct logic diagrams from a boolean expression Be able to construct truth tables Be able to convert between 8 bit binary and <256 denary numbers	Be able to represent an algorithm in a flowchart or pseudocode. Be able to follow through an algorithm in ones head, tracking variables	Be able to convert a simple pseudocode algorithm to python Be able to debug code containing simple syntax errors Be able to use arithmetic operators in code including Modulus and integer division Be able to write programs with inputs and outputs	Be able to use photopea to create bitmaps and vector graphics Be able to make simple keyframe animations Be able to design a project to match a given brief	Be able to design a website for a given purpose from a brief. Be able to make a website using a program like wickeditor, and from scratch using html and css Be able to add navigation to a website and plan how pages link together
Assessment & Educational Visit Opportunities	End of Topic formative	End of Topic Formative	End of Topic Formative	End of Topic Formative	End of Topic Formative	End of Topic Formative
Lesson Sequence	1. Selection 2 2. Iteration 2 3. Subroutines 4. Sensing and Broadcasting 5. Building a game 6. Assessment	1. Thinking in 1s and 0s 2. Thinking with 8 bits 3. Converting from denary 4. Logic gates 5. Truth Tables 6. Assessment	1. Algorithms 2. Computational thinking 3. Representing Algorithms 4. Searching and sorting 1 5. Searching and sorting 2 6. Assessment	1. Your first Python program 2. Debugging 3. Working with numbers 4. Selection 5. Iteration (While) 6. Assessment	1. Bitmap graphics 2. Vector graphics 3. Animation 1 4. Animation 2 5. Project 6. Assessment	1. Creating a webpage with HTML 2. Styling a webpage with CSS 3. Adding images and other content 4. Making website accessible 5. Navigating to other pages 6. Creating a personal website

Computer Science Curriculum Map

Key knowledge & skills to be mastered by students			Year 9			
	Learning Period 1:Autumn	**Learning Period 2:Autumn**	**Learning Period 3: Spring**	**Learning Period 4: Spring**	**Learning Period 5: Summer**	**Learning Period 6: Summer**
Topic title	**9.1 Binary and Data Representation**	**9.2 Programming With Python 2**	**9.6 Computer Systems 2**	**9.4 Cybersecurity**	**9.5 Mobile App Development**	**9.6 Data Science**
Core principle	Understand simple Boolean logic Maximising the use of Horizons	Use a wide range of software and technology Modify and create computer programs Use computational thinking skills to solve real world problems Understand simple Boolean logic Maximising the use of Horizons	Use a wide range of software and technology Recognise computer hardware and understand how each component works Maximising the use of Horizons	Embed fundamental ICT skills Be aware of the risks of technology and how they can be minimised Understand what networks are and how they are used Recognise and predict technology trends Maximising the use of Horizons	Use a wide range of software and technology Modify and create computer programs Create and edit a variety of media Use computational thinking skills to solve real world problems Recognise and predict technology trends Maximising the use of Horizons	Use a wide range of software and technology Embed fundamental ICT skills Be aware of the risks of technology and how they can be minimised Use computational thinking skills to solve real world problems Recognise and predict technology trends Maximising the use of Horizons
Declarative Knowledge (Facts, rules, principles, and the relationships between them)	Knowledge of how binary is used to store numbers Knowledge of how binary is used to store text Knowledge of how binary is used to store audio Knowledge of how binary is used to store images Know the meaning of keywords; sampling, character set, encoding, binary overflow, analogue, digital	Knowledge of how computers generate "pseudorandomness" Knowledge of selection, iteration, subroutine structures in code Know the syntax of python's random library. Knowledge of how to evaluate code in terms of maintainability	Know the purpose and features of the CPU Know the 3 types of secondary storage and how they store bits Know the advantages of the 3 storage technologies Know the difference and similarities between Cache, RAM, and secondary storage Know how computers are connected together and why this is beneficial Know the purpose of the operating syste and some features it provides	Know why cybersecurty is necessary and important. Know the most common forms of cyberattack Know the most common forms of network security Know common network vulnerabilities Know some causes of ethical hacking Know the difference between social engineering and system vulnerabilities	Know the definition and common features+examples of GUIs Know the difference between inputs and outputs from a software perspective Know the main features of event based programming Know the different types of error	Know the definition and importance of big data Know the purpose and types of visualisation Know what can be gained from processing big data Know links between big data, privacy, and AI Know the PPDAC cycle Know the different ways to gather data, primary/secondary sources, surveys and experiments
Procedural Knowledge (Knowledge of methods and processes that can be performed)	Be able to add 8 bit numbers in binary Be able to convert a simple image file to binary Be able to turn an analogue signal into a digital one Be able to translate text between ascii binary and text with a provided character set	Be able to trace the value of paramaters and variables through a program using truth tables Be able to follow the control flow of a medium complex program (one that uses nested structures) Be able to design and write a program to fulfill a simple task	Be able to label a diagram of the computer Be able to choose the best storage technology and justify decision based on a brief Be able to compare the processing speeds of 2 CPUs Be able to draw a network diagram and justify how they've been connected	Be able to match vulnerabilities, prevention methods, and cyberattacks	Be able to find syntax and logic errors in code Be able to move icons on a GUI Be able to design an app with navigation system	Be able to create a visualisation using excel and a given data set Be able to choose the best type of visualisation for a job Be able to draw conclusions from a visualisation and averages
Assessment & Educational Visit Opportunities	End of Topic formative	End of Topic Formative	End of Topic Formative	End of Topic Formative	End of Topic Formative	End of Topic Formative
Lesson Sequence	1. Converting binary 2. Binary addition 3. Data Representation ASCII 4. Data Representation Images 5. Data Representation Sound 6. Assessment	1. Data types 2. Selection 3 3. Iteration 2 4. Iteration 3 5. Randomness in programs 6. Assessment	1. The CPU 2. Memory and Storage 3. Comparing storage devices 4. The Operating System 5. Connecting Systems 6. Assessment	1. Data protection 2. Human Error 3. Ethical and unethical hacking 4. Malware 5. Designing for security 6. Assessment	1. GUI 2. Event driven programming 3. Fixing errors 4. User Input 5. Project 1 6. Project 2	1. Data and visualisations 2. Big data 3. Data trends 4. Interrogating data 5. Cleaning data 6. Assessment

Computer Science Curriculum Map

Key knowledge & skills to be mastered by students				Year 9		
	Learning Period 1:Autumn	**Learning Period 2:Autumn**	**Learning Period 3: Spring**	**Learning Period 4: Spring**	**Learning Period 5: Summer**	**Learning Period 6: Summer**
Topic title	**9.1 Binary and Data Representation**	**8.3 Computational thinking and algorithms**	**9.2 Programming With Python 2**	**9.6 Data Science**	**9.5 Mobile App Development**	**9.4 Cybersecurity**
Core principle	Understand simple Boolean logic Maximising the use of Horizons	Use a wide range of software and technology Embed fundamental ICT skills Use computational thinking skills to solve real world problems Understand simple Boolean logic Maximising the use of Horizons	Use a wide range of software and technology Modify and create computer programs Use computational thinking skills to solve real world problems Understand simple Boolean logic Maximising the use of Horizons	Use a wide range of software and technology Embed fundamental ICT skills Be aware of the risks of technology and how they can be minimised Use computational thinking skills to solve real world problems Recognise and predict technology trends Maximising the use of Horizons	Use a wide range of software and technology Modify and create computer programs Create and edit a variety of media Use computational thinking skills to solve real world problems Recognise and predict technology trends Maximising the use of Horizons	Embed fundamental ICT skills Be aware of the risks of technology and how they can be minimised Understand what networks are and how they are used Recognise and predict technology trends Maximising the use of Horizons
Declarative Knowledge (Facts, rules, principles, and the relationships between them)	Knowledge of how binary is used to store numbers Knowledge of how binary is used to store text Knowledge of how binary is used to store audio Knowledge of how binary is used to store images Know the meaning of keywords; sampling, character set, encoding, binary overflow, analogue, digital	Knowledge of abstraction, algorithmic thinking, decomposition, patternet recognition Knowledge of what each symbol in a flowchart is for Knowledge of pseudocode structures Knowledge of Linear and binary search Knowledge of basic sorting algorithms Knowledge of uses for sorting algorithms	Knowledge of how computers generate "pseudorandomness" Knowledge of selection, iteration, subroutine structures in code Know the syntax of python's random library. Knowledge of how to evaluate code in terms of maintainability	Know the definition and importance of big data Know the purpose and types of visualisation Know what can be gained from processing big data Know links between big data, privacy, and AI Know the PPDAC cycle Know the different ways to gather data, primary/secondary sources, surveys and experiments	Know the definition and common features+examples of GUIs Know the difference between inputs and outputs from a software perspective Know the main features of event based programming Know the different types of error	Know why cybersecurty is necessary and important. Know the most common forms of cyberattack Know the most common forms of network security Know common network vulnerabilities Know some causes of ethical hacking Know the difference between social engineering and system vulnerabilities
Procedural Knowledge (Knowledge of methods and processes that can be performed)	Be able to add 8 bit numbers in binary Be able to convert a simple image file to binary Be able to turn an analogue signal into a digital one Be able to translate text between ascii binary and text with a provided character set	Be able to represent an algorithm in a flowchart or pseudocode. Be able to follow through an algorithm in ones head, tracking variables	Be able to trace the value of paramaters and variables through a program using truth tables Be able to follow the control flow of a medium complex program (one that uses nested structures) Be able to design and write a program to fulfill a simple task	Be able to create a visualisation using excel and a given data set Be able to choose the best type of visualisation for a job Be able to draw conclusions from a visualisation and averages	Be able to find syntax and logic errors in code Be able to move icons on a GUI Be able to design an app with navigation system	Be able to match vulnerabilities, prevention methods, and cyberattacks
Assessment & Educational Visit Opportunities	End of Topic formative	End of Topic Formative	End of Topic Formative	End of Topic Formative	End of Topic Formative	End of Topic Formative
Lesson Sequence	1. Converting binary 2. Binary addition 3. Data Representation ASCII 4. Data Representation Images 5. Data Representation Sound 6. Assessment	1. Algorithms 2. Computational thinking 3. Representing Algorithms 4. Searching and sorting 1 5. Searching and sorting 2 6. Assessment	1. Data types 2. Selection 3 3. Iteration 2 4. Iteration 3 5. Randomness in programs 6. Assessment	1. Data and visualisations 2. Big data 3. Data trends 4. Interrogating data 5. Cleaning data 6. Assessment	1. GUI 2. Event driven programming 3. Fixing errors 4. User Input 5. Project 1 6. Project 2	1. Data protection 2. Human Error 3. Ethical and unethical hacking 4. Malware 5. Designing for security 6. Assessment

Computer Science Curriculum Map

Key knowledge & skills to be mastered by students			Year 10			
	Learning Period 1:Autumn	**Learning Period 2:Autumn**	**Learning Period 3: Spring**	**Learning Period 4: Spring**	**Learning Period 5: Summer**	**Learning Period 6: Summer**
Topic title	**Program Planning + Processing**	**Control Structures + Storage**	**Subroutines + Algorithmics**	**Data Structures + Cybersecurity**	**File Handling + Binary maths**	**Software Development Project + Data Representation**
Core principle	Use a wide range of software and technology Embed fundamental ICT skills Modify and create computer programs Use computational thinking skills to solve real world problems Recognise computer hardware and understand how each component works Recognise and predict technology trends Maximising the use of Horizons	Use a wide range of software and technology Embed fundamental ICT skills Modify and create computer programs Use computational thinking skills to solve real world problems Recognise computer hardware and understand how each component works Understand simple Boolean logic Recognise and predict technology trends Maximising the use of Horizons	Use a wide range of software and technology Modify and create computer programs Use computational thinking skills to solve real world problems Understand simple Boolean logic Maximising the use of Horizons	Use a wide range of software and technology Modify and create computer programs Use computational thinking skills to solve real world problems Understand simple Boolean logic Recognise and predict technology trends Maximising the use of Horizons	Use a wide range of software and technology Modify and create computer programs Be aware of the risks of technology and how they can be minimised Use computational thinking skills to solve real world problems Understand simple Boolean logic Maximising the use of Horizons	Use a wide range of software and technology Embed fundamental ICT skills Modify and create computer programs Create and edit a variety of media Use computational thinking skills to solve real world problems Understand simple Boolean logic Maximising the use of Horizons
Declarative Knowledge (Facts, rules, principles, and the relationships between them)	Know the definition of sequence, variables, debugging, Translators, casting. Know the purpose of different symbols used in flowcharts Know the 4 main variable types Know the 3 types of primary memory and their purpose Know the 4 registers and their purposes Know the 3 factors that affect the CPU's performance Know the purpose differences of RAM and ROM	Know the 7 arithmetic operators and how to use them in python Know the definition and syntax of python selection structures Know the significance to nesting of indentation Know the meaning of iteration and syntax of iteration structures in python Know the purpose and types of data validation Know the 3 logic operators Know how the 3 storage technologies store 0s and 1s Know the advantages and disadvantages of the 3 storage technologies Know the logic gate symbols for each boolean operators	Know the difference between subroutines, procedures, and functions Know the syntax for subroutines in python Know how subroutines make code more maintainable Know the definition of scope and the difference between global and local variables Know the definitions of decomposition, algorithmic thinking, pattern recognition, and decomposition and how they can help approach all problems Know the purpose and layout of a trace table	Know how strings are stored as lists of characters Know how characters are stored as ascii codes Know how to use indexes to get elements out of a list. Know the syntax of working with lists in python Know the different threats online and to computer systems Know the value of data and the importance of cyber-security Know the defences against cybercrime Know the difference between social engineering attacks and threats to systems or networks	Know why programs have to interact with files Know the different ways to open a file, write/read/append Know the syntax for file handling in python Know how numbers are represented in binary, including negatives and hexadecimals Know the effect binary shifts have on the value and accuracy of the number	Know how character set affects the size of a text file Know how audio signals are converted into digital ones Know how souind, bitmaps, and text files are stored in binary Know how quality and file size are affected by how many bits are used for storage Know the different types of compression and the effect on file size and quality
Procedural Knowledge (Knowledge of methods and processes that can be performed)	Be able to order statements to make a program work Be able to track the value of multiple variables through a program Be able to plan algorithms/programs using flowcharts Be able to interpret error messages Be able to fix type errors using casting Be able to talk through the stages a CPU will do given a simpole program	Be able to complete truth tables for a given logic diagram Be able to select a storage type for a given use Be able to combine arithmetic and logic operators to control selection/iteration structures Be able to correctly apply while and for loops when iteration is needed Be able to understand and compare computers based on a list of specs	Be able to fill in a trace table for a given program with some control structures and multiple variables,/parameters Be able to fix indentation scope errors when working with functions Be able to use subroutines in programs to minimise code duplication.	Be able to match vulnerabilities, attacks, and prevention methods Be able to suggest relevant security and risks for a given scenario Be able to write programs with lists Be able to use index reference 2D arrays in python Be able to use for loops to iterate through lists in python Be able to design a program with lists for a given task	Be able to write programs that read and write to files in python Be able to convert numbers between binary, decimal , and hexadecimal Be able to perform binary addition and subtraction and show working Be able to perform left and right binary shifts	Be able to suggest compression type based on a given scenario Be able to convert an analogue signal to a digital one for a given bit depth and sample rate Be able to perform a huffman coding on a short simple string Be able to apply run length encoding to simple pixel art
Assessment & Educational Visit Opportunities	End of Topic formative	End of Topic Formative	End of Topic Formative	End of Topic Formative	End of Topic Formative	End of Topic Formative & Mock Exam
Lesson Sequence	Practical 1. Introduction to Computer Science 2. Translators 3. Sequence 1 4. Sequence 2 5. Variables 6. Input 7. Casting 8. Debugging 9. Flowcharts 1 10. Flowcharts 2 11. Solving complex problems 12. Assessment Theory 1. Computer Systems	Practical 1. Randomisation 2. Arithmetic Expressions 3. Selection 4. Selection Challenge 5. Logical Expressions 6. Nested Selection 7. While Loops 8. Trace Tables 9. For Loops 10. Data Validation 11. Data Validation 2 12. Assessment Theory 1. Optical and Magnetic Storage	Practical 1. Pseudocode 2. Subroutines 1 3. Subroutines 2 4. Functions 5. Functions 2 6. Scope 7. Constants 8. Structured Programming 1 9. Structured Programming 2 10. Creating a larger program 1 11. Creating a larger program 2 12. Assessment Theory 1. Computational thinking	Practical 1. String Handling 1 2. String handling 2 3. String Handling 3 4. ASCII Conversions 5. Programming challenge (Strings) 6. Arrays and Lists 7. List Methods 1 8. List Methods 2 9. 2D Arrays and Lists 10. Programming Challenge (Lists) 11. Programming Challenge (Lists)2 12. Assessment Theory 1. Cybersecurity	Practical 1. Reading text files 2. Writing to text files 3. Working with CSV files 4. Write to CSV files 5. Good programming practices 6. Good programming practices 2 7. Project planning Theory 1. What is representation 2. Number bases (binary to denary) 3. Number bases (denary to binary) 4. Binary addition 5. Binary subtraction 6. Binary shift	Practical 1. Project design 1 2. Project design 2 3. Project building 1 4. Project Building 2 5. Project testing 6. Project Evaluation Theory 1. Unicode and file size calculation 2. Representing bitmap images 3. Bitmap file size calculation 4. Representing sound 5. Sound file size calculation 6. Measurements of storage

Computer Science Curriculum Map

	2. The CPU 3. CPU Registers 4. The FDE Cycle 5. Main Memory 6. Secondary Storage	2. Selecting a storage device 3. Computer specifications 4. Boolean Logic 5. Logic Problems 6. Logic Problems 2	2. Representing Algorithms 3. Tracing Algorithms 1 4. Tracing Algorithms 2 5. Reading algorithms 6. Building algorithms	2. Non-Automated Cybercrime 3. Automated Cybercrime 4. Software Design as a Defence 5. Network Design as a Defence Where is the 6. Danger	7. Signed binary integers 8. Hexadecimal 1 9. Hexadecimal 2 10. Representing Text 11. Assessment	7. Compression 8. Run length encoding 9. Huffman Coding 10. Operating Systems 11. System Software 12. Assessment

Computer Science Curriculum Map

Key knowledge & skills to be mastered by students			Year 11			
	Learning Period 1:Autumn	**Learning Period 2:Autumn**	**Learning Period 3: Spring**	**Learning Period 4: Spring**	**Learning Period 5: Summer**	**Learning Period 6: Summer**
Topic title	**Nesting and Algorithms**	**Data structures + Networks**	**Program design + Ethics**	**Revision**	**Revision**	
Core principle	Use a wide range of software and technology Use computational thinking skills to solve real world problems Understand simple Boolean logic Maximising the use of Horizons	Use a wide range of software and technology Use computational thinking skills to solve real world problems Understand what networks are and how they are used Understand simple Boolean logic Maximising the use of Horizons	Use a wide range of software and technology Be aware of the risks of technology and how they can be minimised Use computational thinking skills to solve real world problems Understand simple Boolean logic Maximising the use of Horizons			
Declarative Knowledge (Facts, rules, principles, and the relationships between them)	Recap programming Know the purpose and process of linear and binary search, and merge, insertion, and bubble sort Know the advantages of different sort algorithms Know how to measure and improve algorithms efficiency	Know the structure and purpose of csv files Know the syntax for interacting with csv files in python Recap python data structures and file handling Know the advantages and disadvantages of wired and wireless networks Know the process by which networks are accessed Know the different pieces of hardware used for networks Know the TCP/IP model and layers of the Internet Know the difference between clien server and peer to peer networks	Know the 3 laws relevant to computing, Data Protection, Computer Misuse, Copyright Know the different ethical legal cultural environmental privacy issues surrounding technology Know how technology has and continues to impact society, including knowledge of some emerging technologies Know some of the penalties for committing cybercrimes			
Procedural Knowledge (Knowledge of methods and processes that can be performed)	Recap Programming Be able to apply search and sort algorithms, showing the steps Be able to write functions for each of the search and sort algorithms	Recap python data structures and file handling Be able to write programs that read and write from csv files Be able to debug file handling errors Be able to suggest a type of network and for a given purpose	Be able to design and write a program to a given brief Be able to discuss relevant ethical legal cultural environmental issues to a given scenario			
Assessment & Educational Visit Opportunities	End of Topic formative	End of Topic Formative & Mock Exam 1	End of Topic Formative	Mock Exam 2		
Lesson Sequence	Practical 1. Sequence 2. Selection 3. Iteration 4. Input and debugging 5. Subroutines 6. String Handling 7. Data Validation Theory 1. Linear Search 2. Binary Search 3. Comparing searching Algorithms 4. Coding searching algorithms 5. Bubble Sort 6. Merge Sort 7. Comparing sort algorithms 8. Coding sorting algorithms 9. Fit for purpose? 10. Improving algorithms 11. Assessment	Practical 1. Arrays and lists 2. 2D arrays and lists 3. Structured programming 4. Working with text files 5. Working with CSV files 6. Good programming practice 7. Debugging Theory 1. What is a computer network 2. The client server Model 3. Network Hardware 4. Network topologies 5. Wired and wireless transmission media 6. Network performance 7. What is the internet 8. Hosting services 9. Protocols 10. The TCP/IP Model 11. Assessment	Practical 1. Creating programs to solve problems 1 2. Creating programs to solve problems 2 3. Creating programs to solve problems 3 4. Creating programs to solve problems 4 5. Creating programs to solve problems 5 6. Creating programs to solve problems 6 Theory 1. The impact of technology 2. The law, data protection 3. The law, Copyright 4. The Computer Misuse Act 5. Cultural Impacts 6. Privacy and Surveillance 7. Environmental impact 8. Ethical Impact 9. Cybercrimes 10. Defending against cybercrimes 11. Consequences of cybercrimes 12. Assessment	Revision	Revision	

Computer Science Curriculum Map

Key knowledge & skills to be mastered by students				Year 11		
	Learning Period 1:Autumn	**Learning Period 2:Autumn**	**Learning Period 3: Spring**	**Learning Period 4: Spring**	**Learning Period 5: Summer**	**Learning Period 6: Summer**
Topic title	2.2 Programming fundamentals - 2.3 producing robust programmes	2.5 programming languages and IDE's - programming project	Revision	**Revision**	**Revision**	
Core principle	Use a wide range of software and technology Modify and create computer programs Use computational thinking skills to solve real world problems Recognise and predict technology trends Understand simple Boolean logic Maximising the use of Horizons	Use a wide range of software and technology Modify and create computer programs Use computational thinking skills to solve real world problems Recognise and predict technology trends Understand simple Boolean logic Maximising the use of Horizons				
Declarative Knowledge (Facts, rules, principles, and the relationships between them)	Knowledge of python syntax Knowledge of what variables are Knowledge of Boolean and comparison operators Knowledge of data types Knowledge of records Knowledge of SQL queries Knowledge of subprogrammes and their importance Knowledge of ways to make a programme robust Knowledge of programme maintainability Knowledge of testing Knowledge of both syntax and logic errors Knowledge of what makes test data suitable Knowledge of how to refine algorithms	Knowledge of different languages and their uses Knowledge of the purpose of translators, interpreters and compilers. Knowledge of the features of an IDE Knowledge of what variables are Knowledge of Boolean and comparison operators Knowledge of data types Knowledge of subprogrammes and their importance				
Procedural Knowledge (Knowledge of methods and processes that can be performed)	Be able to define and change the value in a variable Be able to use Boolean and comparison operators in algorithms Be able to cast and recast data Be able to wright an SQL query Be able to wright a simple programme Be able use subprogrammes to build complex programmes Be able to identify robust programmes Be able to comment code Be able to name variables appropriately Be able to identify syntax errors Be able to identify logic errors	Be able to distinguish between compilers and interpreters Be able to explain how translators work Be able to define and change the value in a variable Be able to use Boolean and comparison operators in algorithms Be able to cast and recast data Be able to wright a simple programme Be able use subprogrammes to build complex programmes				
Assessment & Educational Visit Opportunities	End of Topic formative	End of Topic Formative & Mock Exam 1		Mock Exam 2		
Lesson Sequence	1. programming constructs 2. comparison operators 3. Boolean operators 4. data types and casting 5. string manipulation 6. file handling 7. records 8. SQL 9. Subprograms	1. characteristics and purpose of languages 2. translators 3. compilers and interpreters 4. IDE's 5. Test. 6. Programming project 7. Programming project 8. Programming project 9. Programming project	Revision	Revision	Revision	

	10. Arrays 11. Random number generator 12. 2.2 Test 13. defensive design considerations 14. defensive design considerations 2 15. maintainability 16. types of testing 17. syntax and logic errors 18. suitable test data 19 refining algorithms 20. Test	10. Programming project 11. Programming project 12. Programming project 13. Revision 14. Revision 15. Revision 16. Revision 17. Revision 18. Revision				