The Royal School

Wolverhampton

Curriculum 2024 – 2025

Computer Science and IT

Curriculum 2024 - 2025 Subject Computer Science

Computer Science and IT curriculum intent

Whilst following the guidelines of the National Curriculum we aim to offer a broad based curriculum which will result in acquisition of knowledge, develop enjoyment of learning and will equip our pupils for work and leisure as active, confident and responsible members of society.

In particular we aim to:

- To become 'tech literate' and use technology responsibly and creatively
- To know how to use technology and the Internet safely
- To understand that data can be presented in different ways and manipulated
- To write algorithms to create a process
- To test their algorithms and debug them

Computer Science and IT curriculum implementation

In Computer Science and IT we teach 2 lessons per week at KS3, 3 lessons per week at KS4 and 5/6 lessons per week at KS5. Our curriculum is structured in learning cycles. Each cycle lasts for 7 weeks and includes at least one assessment followed by a review where re-teaching or stretch and challenge opportunities, tailored to the needs of the pupils can take place. There are 5 learning cycles per year.

Each lesson follows The Royal lesson structure below:

- Date and learning question.
- Review questions as bell work and answers to be self-assessed or peer assessed.
- Homework set at start of lesson.
- The learning journey shared including lesson objectives and success criteria.
- Challenge tasks set every lesson.
- Review learning objectives at end of lesson.

Our curriculum is implemented in many ways including taught lessons and out of class experiences such as:

- Using Office 365 features including, OneNote, Teams and collaboration documents to continue learning outside the classroom.
- Digital media and 3D design Lesson during the Lesson 7 extended day. Covering additional fun and engaging activities that curriculum time does not always allow.
- Use of Nearpod, Socratives and other online platforms, this allows for engagement and to allow students to use technology within learning.

Computer Science and IT curriculum impact

The impact of our curriculum can be evaluated in many ways using both quantitative and qualitiative information indicating how ready pupils are for the next stage in their learning whether that be transition between key stages or leaving for universities, apprenticeships or work at the end of year 13.

This includes an assessment of

- the number of pupils achieving the national average (and often higher!) at the end of their key stage indicating their readiness to move forward with the next stage of their learning journey.
- in class assignments and coursework
- learning cycle tests and mock exams
- assessment of learning within the classroom
- assessing group work and projects

Further information that can be used to assess the impact of the curriculum includes:

- The number of pupils gaining places at their first choice universities.
- The number of pupils gaining entry into their chosen career route including apprenticeships and employment.
- The number of students who take Computer Science or ICT at Key Stage 4 or Key Stage 5.

The Royal School

Computer Science and IT Learning Journey

Overview of the curriculum

Primary School -
Barefoot Curriculum
Cornerstones Curriculum

KS3
eSafety, Web Design,
Visual Programming, Computing Theory

KS4
GCSE Computer Science 9-1
BTEC ICT Level 2 / Cambridge Techincals Level 2

KS5
A Level Computer Science
Cambridge Techincals Level 3 IT

Final Steps
Degree Based Apprenticeships
University Degree
Employment

OCR
RECOGNISING ACHIEVEMENT

Pearson

The Royal School

Computer Science and IT Learning Journey

The journey explained

Primary School

Computer Science is not a discrete subject but embedded into a wider curriculum. In KS1, this will be using technology in a lesson. This could be as simple as using the internet for research or using a device like an iPad or camera. Computer Science to facilitate the learning journey but not to create it.

Moving into KS2, this is when concepts such as 'sequencing' come into and students will start to focus on steps needed for Computational Methods. Such as drawing shapes, programming robots to move around a room and working with sensors (lights, sound and movement).

Senior School – KS3

KS3 is where Computer Science is taught as its own subject for the first time. However, it falls under the Technology umbrella with Food Technology and Design Technology. Students in Year 7 and 8 receive 12 weeks of Computer Science each Year.

Therefore, the key principles focused on are:

- Sequence
- Selection
- Iteration
- E Safety

In Year 7, this will be focused on by learning HTML and CSS. This is the simplest scripting language to learn, as there are no syntax errors. If code is wrote incorrectly, it just does not work. Students get an understanding of a text based input that produces a visual output in the form of a website.

This transitions nicely into Year 8 where students begin to use Scratch, a visual programming tool. They will begin to look at entry-level GCSE problems and solve them visually. Scratch is based like a jigsaw puzzle; therefore, if the blocks do not fit they cannot be used. This allows students to make much more complex programs than they could in HTML.

Senior School – KS4

Students now have the option to choose GCSE Computer Science or BTEC IT. The KS3 curriculum is based on Computer Science, however, student will also use many IT skills such as:

- Word Processing
- Data Entry
- Presentation skills
- File management

GCSE Computer Science consists of two examination Units. Unit 1 is Computer Theory; most of these concepts will be new to students, as Computer Theory not covered in detail at KS3. This also helps if students join the school during KS4 as they can gain the knowledge alongside the other students. Unit 2 is the Algorithms and Programming; students will now be able to translate the visual algorithms from KS3 into text-based algorithms in Python.

BTEC ICT has three units. 60% coursework and 40% exam. Unit 1 is creating a user interface. Unit 2 is Managing Data. Unit 1 and Unit 2 are the coursework units. Unit 3 is the written examination unit. Students will be able to take the basic techniques learned from using IT at KS3 and apply them to real problems. Creating user friendly interfaces and simulating how business would use and manage data.

Cambridge Nationals has three units. Unit R050: IT in the digital world, Unit R060: Data manipulation using spreadsheets, Unit R070: Using Augmented Reality to present information Unit R060: Data manipulation using spreadsheets, 60% coursework and 40% exam.

Senior School KS5

We over the same courses at GCSE as we do at A Level and Level 3. If students continue with us from KS4 to KS5 they will have perfect continuity. The courses are structured the same but just to a higher level, expanding upon KS4 knowledge.

Final Steps

Students can move onto Degree Based Apprenticeships or University Degrees within IT and Computer Science

Potential careers or Employment in:

- Software developer
- Database administrator
- Computer hardware engineer
- Computer system analyst
- Computer network architect
- Computer and information research scientist
- Computer information systems manager
- Project manager in IT computer science
- Web developer

The Royal School

SUBJECT KS3 Curriculum Mapping

https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/840002/Secondary_national_curriculum_corrected_PDF.pdf

National curriculum content	Year 7	Year 8	Year 9 (GCSE foundation year)
Aims:			
The national curriculum for computing aims to ensure that all pupils:			
• can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation	[not covered]	[partially covered – Visual Programming]	[Unit 2 - covered]
• can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems	[not covered]	[covered Game Development]	[Unit 2 - covered]
• can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems are responsible, competent, confident and creative users of information and communication technology	[partially covered e-safety]	[partially covered Visual Programming]	[Unit 1 - covered]
Subject Content:			
• design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems	[partially covered Web Development]	[covered Visual Programming]	[Unit 1 - covered]
• understand several key algorithms that reflect computational thinking [for example, ones for sorting and searching]; use logical reasoning to compare the utility of alternative algorithms for the same problem	[not covered Web Development]	[partially covered Visual Programming]	[Unit 2 - covered]
• use 2 or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data	[partially covered Web Development]	[partially covered Visual Programming]	[Unit 2 - covered]

structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions • understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in circuits and programming; understand how numbers can be represented in binary, and be able to carry out simple operations on binary numbers [for example, binary addition, and conversion between binary and decimal]	[not covered]	[partially covered Visual Programming]	[Unit 2 - covered]
• understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems	[partially covered e-safety]	[partially covered Game Design]	[Unit 1 - covered]
• understand how instructions are stored and executed within a computer system; understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits	[not covered]	[partially covered Visual Programming]	[Unit 2 - covered]
• undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users	[partially covered e-safety]	[partially covered Game Design]	[Unit 1 - covered]
• create, reuse, revise and repurpose digital artefacts for a given audience, with attention to trustworthiness, design and usability	[partially covered Web Development]	[partially covered Visual Programming]	[Unit 2 - covered]
• understand a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct, and know how to report concerns	[covered e-safety]	[covered e-safety recap]	[Unit 1 - covered]

The Royal School

Computer Science Curriculum Map and Assessment 2024-25

	Year 7	Year 8	Year 9 1 x theory 1 x programming	Year 10 1 x theory 1 x programming	Year 11	Year 12	Year 13
Learning Cycle 1	• E safety • Digital Literacy • Word Processing • Presentation software • Graphic Design • Spreadsheet and databases ○ On rotation	• Computational thinking • Problem solving • Visual programming • Text based programming • Algorithms and flowcharts ○ On rotation	1.1.1 CPU Architecture 1.1.2 CPU Performance 1.1.3 Embedded [6 weeks] 2.2.1 Programming fundamentals and 2.2.2 Datatypes [6 weeks]	1.2.5 Compression [6 weeks] 2.1.1 Computational thinking [6 weeks]	Exam Practice and revision	1.1.1 Structure and function of the processor 1.1.2 Types of processor 1.1.3 Input, output and storage 2.3.1 Algorithms	1.2.3 Software Development 1.3.2 Databases Unit 2 Coursework
Learning Cycle 2	• E safety • Digital Literacy • Word Processing • Presentation software • Graphic Design • Spreadsheet and databases On rotation	• Computational thinking • Problem solving • Visual programming • Text based programming • Algorithms and flowcharts On rotation	1.5.1 Operating systems 1.5.2 Utility software [6 weeks] 2.2.1 Programming fundamentals and 2.2.2 Datatypes [6 weeks]	1.3.1 Network topologies [6 weeks] 2.1.2 Design, create and refining algorithms [6 weeks]	Exam Practice and revision	1.2.4 Types of Programming Language 1.4.1 Data Types 2.2.1 Programming techniques	1.3.3 Networks 1.3.4 Web Technologies Unit 2 Coursework
Learning Cycle 3	• E safety • Digital Literacy • Word Processing • Presentation software • Graphic Design	• Computational thinking • Problem solving • Visual programming • Text based programming	1.2.1 Primary memory 1.2.2 Secondary storage [6 weeks] 2.2.3 Additional	1.3.2 Wired and wireless networks, protocols and layers [6 weeks] 2.1.3 Searching and	Exam Practice and revision	1.2.1 Systems Software 1.2.2 Applications Generation 2.2.2 Computati	1.4.3 Boolean Algebra Unit 2 Revision and practice

	• Spreadsheet and databases On rotation	• Algorithms and flowcharts On rotation	programming techniques [6 weeks]	sorting algorithm [6 weeks]		onal Thinking methods	
Learning Cycle 4	• E safety • Digital Literacy • Word Processing • Presentation software • Graphic Design • Spreadsheet and databases On rotation	• Computational thinking • Problem solving • Visual programming • Text based programming • Algorithms and flowcharts On rotation	1.2.3 Units 1.2.4 Data storage [6 weeks] 2.4.1 Boolean Logic [6 weeks]	1.4.1 Network Security 1.4.2 Identifying and preventing vulnerabilities [6 weeks] 2.3.1 Defensive design 2.3.2 Testing [6 weeks]	Exam Practice and revision	1.3.1 Compression, Encryption and Hashing 1.4.2 Data Structures Unit 2 Revision and practice	Final exam prep and revision
Learning Cycle 5	• E safety • Digital Literacy • Word Processing • Presentation software • Graphic Design • Spreadsheet and databases On rotation	• Computational thinking • Problem solving • Visual programming • Text based programming • Algorithms and flowcharts On rotation	Producing robust programs Assessment and Project [6 weeks]	1.6.1 Ethical legal, cultural , environment impacts of digital technology [6 weeks] 2.5.1 Languages 2.5.2 IDE [6 weeks]	GCSE/BTEC exams followed by A level bridging work	1.5.1 Computing related legislation 1.5.2 Moral and ethical Issues Unit 2 Coursework	A level/BTEC exams followed by University bridging work

NB. The detailed SOW, assessments and lessons for each learning cycle should be placed in the correct folder in the subject team

The Royal School

IT Curriculum Map and Assessment 2024-25

	Year 9 BTEC IT	Year 10 BTEC IT	Year 11 BTEC IT	Year 12 OCR Cambridge	Year 13 OCR Cambridge
Learning Cycle 1	Introduction to Unit 1 – Design Principes	Unit 1 Mock Assignment	Exam Preparation	Unit 1 Fundementals of IT	Unit 8 and 9 combined coursework Project management and project development Unit 3 Cyber Security
Learning Cycle 2	Unit 1 Software skills	Unit 1 – Controlled Assessment	Exam preparation	Unit 1 Fundementals of IT	Unit 8 and 9 combined coursework Project management and project development Unit 3 Cyber Security
Learning Cycle 3	Unit 1 Software skills	Unit 2 – Mock Assessment	January external exam Redo either Unit 1 or Unit 2 Controlled assessment	Unit 1 external exam Start Unit 2 Global information	Unit 3 Cyber security external exam Unit 8 and 9 combined coursework Project management and project development
Learning Cycle 4	Unit 2: Data manipulation using spreadsheets	Unit 2 Controlled Assessment	Prepare for Unit 3 Retake / Summer Exam	Unit 2 Global information	Resists and finish Unit 8 and 9 combined coursework Project management and project development

Learning Cycle 5	Unit 2: Data manipulation using spreadsheets	Prepare for Unit 3 Exam	Prepare for Unit 3 Retake / Summer Exam	Unit 2 Global information external exam Start Unit 3 Cyber Security	A level/BTEC exams followed by University bridging work

The Royal School

Computer Science Assessment Calendar 2024-25

	Year 7	Year 8	Year 9	Year 10	Year 11	Year 12	Year 13
Learning Cycle 1	Week 6, Week 10 [rotation]		Week 7	Week 6	Week 5 NEA(coursework) will be collected and assessed every 3 weeks.	Week 3 Week 6	Coursework will be collected and assessed every 3 weeks. Week 3
Learning Cycle 2			Week 15	Week 10	Week 12	Week 14	Week 9 Week 13 Week 16
Learning Cycle 3			Week 16	Week 16	Continual examination preparation assessment.	Week 21	Week 26
Learning Cycle 4			Week 22	Week 25	Week 24	Week 22	Week28
Learning Cycle 5			Week 33	Week 33	GCSE/BTEC exams	Week 33	A level/BTEC exams

NB. The assessments and mark scheme for each learning cycle should be placed in the correct folder in the subject team.

The Royal School

IT Assessment Calendar 2024-25

	Year 9	Year 10	Year 12	Year 13
Learning Cycle 1	Week 6	Week 5	Week 3 Week 6	Coursework will be collected and assessed every 3 weeks. Week 3
Learning Cycle 2	Week 12	Week 11	Week 14	Week 9 Week 13 Week 16
Learning Cycle 3	Week 18	Week 16	Week 21	Week 26
Learning Cycle 4	Week 24	Week 25	Week 22	Week28
Learning Cycle 5	Week 33	Week 33	Week 33	A level/BTEC exams

NB. The assessments and mark scheme for each learning cycle should be placed in the correct folder in the subject team.

KS3 Assessment criteria

Level	Algorithms	Programming & Development	Data & Data Representation	Hardware & Processing	Communication & Networks	Information Technology
	WB – Working below end of year expectations	WT – Working towards end of year expectations	WAT – Working at end of year expectations		WA – Working above end of year expectations	
YEAR 9	Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. Understands that computers need precise instructions. Demonstrates care and precision to avoid errors.	Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text e.g. programmable robots etc. Executes, checks and changes programs. Understands that programs execute by following precise instructions.	Recognises that digital content can be represented in many forms. Distinguishes between some of these forms and can explain the different ways that they communicate information.	Understands that computers have no intelligence and that computers can do nothing unless a program is executed. Recognises that all software executed on digital devices is programmed.	Obtains content from the world wide web using a web browser. Understands the importance of communicating safely and respectfully online, and the need for keeping personal information prive. Knows what to do when concerned about content or being contacted.	Uses software under the control of the teacher to create, store and edit digital content using appropriate file and folder names. Understands that people interact with computers. Shares their use of technology in school. Knows common uses of information technology beyond the classroom. Talks about their work and makes changes to improve it.
YEAR 9	Understands that algorithms are implemented on digital devices as programs. Designs simple algorithms	Uses arithmetic operators, if statements, and loops, within programs. Uses logical reasoning to	Recognises different types of data: text, number. Appreciates that programs can work with different types	Recognises that a range of digital devices can be considered a computer. Recognises and can	Navigates the web and can carry out simple web searches to collect digital content. Demonstrates use	Uses technology with increasing independence to purposefully organise digital content. Shows an

	using loops, and selection i.e. if statements. Uses logical reasoning to predict outcomes. Detects and corrects errors i.e. debugging, in algorithms.	predict the behaviour of programs. Detects and corrects simple semantic errors i.e. debugging, in programs.	of data. Recognises that data can be structured in tables to make it useful.	use a range of input and output devices. Understands how programs specify the function of a general purpose computer.	of computers safely and responsibly, knowing a range of ways to report unacceptable content and contact when online.	awareness for the quality of digital content collected. Uses a variety of software to manipulate and present digital content: data and information. Shares their experiences of technology in school and beyond the classroom. Talks about their work and makes improvements to solutions based on feedback received.
YEAR 9	Designs solutions (algorithms) that use repetition and two-way selection i.e. if, then and else. Uses diagrams to express solutions. Uses logical reasoning to predict outputs, showing an awareness of inputs.	Creates programs that implement algorithms to achieve given goals. Declares and assigns variables. Uses post-tested loop e.g. 'until', and a sequence of selection statements in programs, including an if, then and else statement.variables. Uses post-tested loop e.g. 'until', and a sequence of selection statements in programs, including	Understands the difference between data and information. Knows why sorting data in a flat file can improve searching for information. Uses filters or can perform single criteria searches for information.	Knows that computers collect data from various input devices, including sensors and application software. Understands the difference between hardware and application software, and their roles within a computer system.	Understands the difference between the internet and internet service e.g. world wide web. Shows an awareness of, and can use a range of internet services e.g. VOIP. Recognises what is acceptable and unacceptable behaviour when using technologies and online services.	Collects, organises and presents data and information in digital content. Creates digital content to achieve a given goal through combining software packages and internet services to communicate with a wider audience e.g. blogging. Makes appropriate improvements to solutions based on feedback received, and can comment on the success of the solution.

		an if, then and else statement				
YEAR 8	Shows an awareness of tasks best completed by humans or computers. Designs solutions by decomposing a problem and creates a sub-solution for each of these parts (decomposition). Recognises that different solutions exist for the same problem.	Understands the difference between, and appropriately uses if and if, then and else statements. Uses a variable and relational operators within a loop to govern termination. Designs, writes and debugs modular programs using procedures. Knows that a procedure can be used to hide the detail with sub-solution (procedural abstraction).	Performs more complex searches for information e.g. using Boolean and relational operators. Analyses and evaluates data and information, and recognises that poor quality data leads to unreliable results, and inaccurate conclusions.	Understands why and when computers are used. Understands the main functions of the operating system. Knows the difference between physical, wireless and mobile networks.	Understands how to effectively use search engines, and knows how search results are selected, including that search engines use 'web crawler programs'. Selects, combines and uses internet services. Demonstrates responsible use of technologies and online services, and knows a range of ways to report concerns.	Makes judgements about digital content when evaluating and repurposing it for a given audience. Recognises the audience when designing and creating digital content. Understands the potential of information technology for collaboration when computers are networked. Uses criteria to evaluate the quality of solutions, can identify improvements making some refinements to the solution, and future solutions.
YEAR 8	Understands that iteration is the repetition of a process such as a loop. Recognises that different algorithms exist for the same problem. Represents	Understands that programming bridges the gap between algorithmic solutions and computers. Has practical experience of a high-level textual language,	Knows that digital computers use binary to represent all data. Understands how bit patterns represent numbers and images. Knows that computers	Recognises and understands the function of the main internal parts of basic computer architecture. Understands the concepts behind the fetch-execute cycle.	Understands how search engines rank search results. Understands how to construct static web pages using HTML and CSS. Understands data transmission	Evaluates the appropriateness of digital devices, internet services and application software to achieve given goals. Recognises ethical issues surrounding

	solutions using a structured notation. Can identify similarities and differences in situations and can use these to solve problems (pattern recognition).	including using standard libraries when programming. Uses a range of operators and expressions e.g. Boolean, and applies them in the context of program control. Selects the appropriate data types.	transfer data in binary. Understands the relationship between binary and file size (uncompressed). Defines data types: real numbers and Boolean. Queries data on one table using a typical query language.	Knows that there is a range of operating systems and application software for the same hardware.	between digital computers over networks, including the internet i.e. IP addresses and packet switching.	the application of information technology beyond school. Designs criteria to critically evaluate the quality of solutions, uses the criteria to identify improvements and can make appropriate refinements to the solution.
YEAR 8	Understands a recursive solution to a problem repeatedly applies the same solution to smaller instances of the problem. Recognises that some problems share the same characteristics and use the same algorithm to solve both (generalisation). Understands the notion of performance for algorithms and appreciates that some algorithms have different performance	Uses nested selection statements. Appreciates the need for, and writes, custom functions including use of parameters. Knows the difference between, and uses appropriately, procedures and functions. Understands and uses negation with operators. Uses and manipulates one dimensional data structures. Detects and corrects syntactical errors.	Understands how numbers, images, sounds and character sets use the same bit patterns. Performs simple operations using bit patterns e.g. binary addition. Understands the relationship between resolution and colour depth, including the effect on file size. Distinguishes between data used in a simple program (a variable) and the storage structure for that data.	Understands the von Neumann architecture in relation to the fetch-execute cycle, including how data is stored in memory. Understands the basic function and operation of location addressable memory.	Knows the names of hardware e.g. hubs, routers, switches, and the names of protocols e.g. SMTP, iMAP, POP, FTP, TCP/IP, associated with networking computer systems. Uses technologies and online services securely, and knows how to identify and report inappropriate conduct.	Justifies the choice of and independently combines and uses multiple digital devices, internet services and application software to achieve given goals. Evaluates the trustworthiness of digital content and considers the usability of visual design features when designing and creating digital artifacts for a known audience. Identifies and explains how the use of technology can impact on society. Designs

	characteristics for the same task.					criteria for users to evaluate the quality of solutions, uses the feedback from the users to identify improvements and can make appropriate refinements to the solution.
YEAR 7	Recognises that the design of an algorithm is distinct from its expression in a programming language (which will depend on the programming constructs available). Evaluates the effectiveness of algorithms and models for similar problems. Recognises where information can be filtered out in generalising problem solutions (abstraction). Uses logical reasoning to explain how an algorithm works. Represents algorithms using structured language	Appreciates the effect of the scope of a variable e.g. a local variable can't be accessed from outside its function. Understands and applies parameter passing. Understands the difference between, and uses, both pre-tested e.g. 'while', and post-tested e.g. 'until' loops. Applies a modular approach to error detection and correction.	Knows the relationship between data representation and data quality. Understands the relationship between binary and electrical circuits, including Boolean logic. Understands how and why values are data typed in many different languages when manipulated within programs.	Knows that processors have instruction sets and that these relate to low-level instructions carried out by a computer.	Knows the purpose of the hardware and protocols associated with networking computer systems. Understands the client-server model including how dynamic web pages use server-side scripting and that web servers process and store data entered by users. Recognises that persistence of data on the internet requires careful protection of online identity and privacy.	Undertakes creative projects that collect, analyse, and evaluate data to meet the needs of a known user group. Effectively designs and creates digital artefacts for a wider or remote audience. Considers the properties of media when importing them into digital artefacts. Documents user feedback, the improvements identified and the refinements made to the solution. Explains and justifies how the use of technology impacts on society, from the perspective of social, economical,

						political, legal, ethical and moral issues.
YEAR 7	Designs a solution to a problem that depends on solutions to smaller instances of the same problem (recursion). Understands that some problems cannot be solved computationally.	Designs and writes nested modular programs that enforce reusability utilising sub-routines where ever possible. Understands the difference between 'While' loop and 'For' loop, which uses a loop counter. Understands and uses two dimensional data structures.	Performs operations using bit patterns e.g. conversion between binary and hexadecimal, binary subtraction etc. Understands and can explain the need for data compression, and performs simple compression methods. Knows what a relational database is, and understands the benefits of storing data in multiple tables.	Has practical experience of a small (hypothetical) low level programming language. Understands and can explain Moore's Law. Understands and can explain multitasking by computers.	Understands the hardware associated with networking computer systems, including WANs and LANs, understands their purpose and how they work, including MAC addresses.	Understands the ethical issues surrounding the application of information technology, and the existence of legal frameworks governing its use e.g. Data Protection Act, Computer Misuse Act, Copyright etc.