Quwwat-Ul-Islam Girls' School



Computing

Knowledge and Skills Progression Map

School Motto
Together we achieve more
School Values
Respect . Aspiration . Responsibility
Learning Drivers
Caring for Ourselves, Our World, Our Education, Our Rights

School Vision Statement

To establish a school of excellence both **Islamically** and **Academically** - Provide a **holistic** education which is inspired by:

Taqwa: God consciousness **morals and values** inspired by the prophet (SAW) coupled with **academic excellence.**

At Quwwat-Ul-Islam girls' school we adhere to the following:

- **Mutual respect and tolerance -** We respect and care for everybody.
- **Democracy -** We promote the right of every individual to have their opinions and voices heard.
- **The rule of the law -** We understand that laws and rules protect us and at the same time provide us with responsibilities, by breaking them there are consequences.

National Curriculum Expectations

EYFS

Understanding the World (Technology)

- Children recognise that a range of technology is used in places such as homes and schools.
- They select and use technology for particular purposes.

Primary Link

Key Stage 1 National Curriculum Expectations
Pupils should be taught:

- understand what algorithms are;
- how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions; create and debug simple programs;
- use logical reasoning to predict the behaviour of simple programs;
- use technology purposefully to create, organise, store, manipulate and retrieve digital content; recognise common uses of information technology beyond school;
- use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the
- internet or other online technologies.

Key Stage 2 National Curriculum Expectations Pupils should be taught:

- design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts; use sequence, selection, and repetition in programs; work with variables and various forms of input and output;
- use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs;
- understand computer networks including the internet; how they provide multiple services, such as the world wide web, and the opportunities they offer for communication and collaboration;
- use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content;
- select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information;
- use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.

Secondary Link:

Key Stage 3 National Curriculum Expectations

Pupils should be taught:

- design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems
- 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
- use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data 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 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]
- understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems
- understand how instructions are stored and executed within a computer system; understand how data of various types (including text, sounds and pictures) be represented and manipulated digitally, in the form of binary digits
- 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
- create, re-use, revise and re-purpose digital artefacts for a given audience, with attention to trustworthiness, design and usability
- 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.

Key Stage 4 National Curriculum Expectations

All pupils must have the opportunity to study aspects of information technology and computer science at sufficient depth to allow them to progress to higher levels of study or to a professional career.

Pupils should be taught:

- develop their capability, creativity and knowledge in computer science, digital media and information technology develop and apply their analytic, problem-solving, design, and computational thinking skills
- understand how changes in technology affect safety, including new ways to protect their online privacy and identity, and how to identify and report a range of concerns.

Computing						
Year Group	**Autumn 1.1**	**Autumn 1.2**	**Spring 2.1**	**Spring 2.2**	**Summer 3.1**	**Summer 3.2**
Year 1	Digital painting	Digital painting	Digital writing	Digital writing	Creating animations in programs	Creating animations in programs
Year 2	Information technology in the world beyond school	Information technology in the world beyond school	Using IT to organise and present data	Using IT to organise and present data	Building sequences in programs	Building sequences in programs
Year 3	Computer networks	Stop-frame animation	Programming sequences using sound	Organising data using databases	Desktop publishing	Events and actions in programs
Year 4	The internet	Audio production	Repetition in programs	Data logging	Photo editing	Using repetition in programming to create a game
Year 5	Introduction to computer systems	Video production	Exploring selection in physical computing	Flat-file databases	Introduction to vector graphics	Using selection in programming to develop a quiz
Year 6	Communication and the internet	Web page creation	Using variables in programming to develop a game	Introduction to spreadsheets	3D modelling	Sensing movement with physical computing
Year 7	Computer networks and data transmission	Using fundamental programming constructs in a block-based language	Physical computing using the micro:bit	Data modelling	Clear messaging in digital media	Using media to gain support for a cause
Year 8	Computer systems and data science	Data representation: text and numbers	Developing vector graphics	Developing for the Web	Mobile app development	Introduction to Python programming
Year 9	Data representation: images and sound	Using data science	Machine learning using the micro:bit	Python programming with sequences of data	Introduction to cybersecurity	3D animation
Year 10 GCSE OCR	Computer Systems	Programming: sequence	Programming: selection	Programming: iteration	Programming: subprograms	Programming: strings and lists
Year 11 GCSE OCR	Programming: dictionaries and data files	Python programming project	HTML programming project	Databases and SQL		

Intent

Computer Science Intent – EYFS to KS4

At QUIG, our aim is to provide an inclusive and ambitious Computer Science curriculum that develops pupils into confident computational thinkers, problem-solvers and programmers.

From EYFS, pupils begin to develop computational thinking through sequencing, patterns, instructions and problem-solving. They use programmable toys and simple digital tools to explore how instructions can control technology and begin to understand algorithms and debugging.

At KS1 and KS2, pupils build on these skills by learning to create, test and debug programs. They develop an understanding of sequence, selection, repetition, algorithms and logical reasoning, while learning how computers, data and networks work.

At KS3, pupils develop more advanced programming and computational thinking skills. They learn to design, write, test and improve programs, while developing their understanding of algorithms, data, computer systems, networks and cybersecurity. Pupils are encouraged to break problems into smaller parts and use logical thinking to find solutions.

At KS4, pupils build on their previous knowledge and prepare for GCSE Computer Science. They develop a deeper understanding of programming, algorithms, data representation, computer systems, networks and cybersecurity. They apply computational thinking to solve increasingly complex problems and evaluate their solutions.

Across EYFS to KS4, our curriculum aims to ensure that pupils:

- **understand how computers work;**
- **develop strong problem-solving and computational thinking skills;**
- **learn to design, write, test and debug programs;**
- **understand algorithms, data, networks and computer systems;**
- **become resilient and independent when solving problems;**
- **use logical thinking to develop effective solutions;**
- **develop the knowledge and skills needed for GCSE Computer Science and future study.**

Ultimately, we want our pupils to move from being users of technology to creators and problem-solvers, with the confidence and skills to understand and shape the digital world around them.

Implementation

The QUIGs Computing curriculum recognises that Computing can be taught both as a discrete subject and through meaningful cross-curricular opportunities. Computing skills are transferable and are therefore applied across subjects such as Art, History, Mathematics, Geography, Science and Literacy. For example, pupils may research the Romans using the internet, evaluate information and present their findings using digital presentations.
Computing is taught discretely once a week, with opportunities to revisit and apply Computing skills regularly across other areas of the curriculum. This approach allows pupils to develop secure knowledge and then apply it in meaningful contexts.
Our curriculum follows the National Curriculum for Computing and develops knowledge and skills across three key areas:

- **Computer Science –** developing computational thinking, algorithms, coding, programming and understanding how computer systems and networks work.
- **Information Technology –** using digital tools to create, store, manipulate, retrieve and present information, including word processing, presentations and data handling.
- **Digital Literacy –** developing pupils' ability to use technology safely, responsibly and respectfully, while evaluating digital content and understanding online risks.

There is a clear progression of knowledge and skills from EYFS to KS4, with new learning building on previously taught concepts. Key knowledge and vocabulary are revisited regularly to support retention and ensure that pupils can confidently use subject-specific terminology.
Vocabulary and oracy are central to our Computing curriculum. Key Computing vocabulary is explicitly introduced, modelled and revisited within lessons. Pupils are given regular opportunities to explain their thinking, describe algorithms, discuss problems and justify their solutions using appropriate technical language. Key vocabulary is displayed in classrooms and included in supporting resources so that pupils can continually revisit and apply new terminology.
Lessons provide opportunities for pupils to explore, practise, apply and evaluate their learning. Programming tasks encourage pupils to test, debug and improve their solutions, developing resilience and problem-solving skills. Teachers use questioning, modelling and practical activities to identify misconceptions and ensure that pupils develop secure understanding.

Through this approach, pupils develop the knowledge, skills and confidence to use technology effectively, think computationally, solve problems and become responsible digital citizens.

Impact

Our Computing curriculum aims to ensure that pupils enjoy Computing, understand its purpose and recognise its importance in their everyday lives and future careers. Pupils develop the knowledge, skills and confidence to use technology creatively, safely and effectively.
By the end of KS4, pupils will:

- Know and remember key Computing knowledge and vocabulary across Computer Science, Information Technology and Digital Literacy.
- Be able to apply their Computing skills independently across Computing and other curriculum areas.
- Demonstrate computational thinking, logical reasoning and problem-solving skills.
- Be able to create, test and debug programs and explain their thinking using appropriate Computing vocabulary.
- Understand how to use technology safely, responsibly and respectfully, including how to recognise risks and where to seek help and support.
- Be confident in using a range of digital tools to research, create, communicate and present information.
- Understand the relevance of Computing to the wider world, creative and technological industries and future career opportunities.
- Develop resilience and independence when solving problems, recognising that mistakes and debugging are an important part of learning.

Measuring Impact

Progress and attainment in Computing are monitored through a range of evidence, including:

- Reviewing pupils' work over time to identify progression in knowledge and skills.
- Observing pupils during lessons to assess their ability to apply knowledge, solve problems and work independently.
- Pupil voice and discussion, allowing pupils to explain what they have learned, use key vocabulary and demonstrate their understanding.
- Assessment and retrieval activities to check pupils' knowledge, understanding and retention of key concepts.

The impact of the curriculum is evident when pupils can explain what they have learned, apply it in new contexts and work increasingly independently. Pupils leave QUIG with the confidence, knowledge and skills to become responsible digital citizens and capable computational thinkers, prepared for the next stage of their education and the wider world.

Computer Science
Knowledge and Skills Progression Map

		Autumn 1.1	Autumn 1.2	Spring 2.1	Spring 2.2	Summer 3.1	Summer 3.2
Year 1	Knowledge	**Digital painting**	**Digital painting**	**Digital writing**	**Digital writing**	**Creating animations in programs**	**Creating animations in programs**
	Skills I can… Key Vocabulary: Digital Art Digital Writing Animation Photography Graphics Audio Video Presentations Data Programming Algorithms Robotics Coding Internet Networks Cybersecurity Online Safety Digital Citizenship Research Communication	Use digital tools to draw pictures. Use paintbrush, eraser and fill tools. Use line and shape tools. Combine shapes, lines and colours to create artwork. Choose appropriate digital tools for different tasks. Use different colours and palettes.	Create artwork inspired by artists such as Kandinsky. Change brush size/style to produce different marks. Use mouse or touchscreen techniques such as clicking, tapping and dragging. Use undo to correct mistakes. Compare creating art digitally with painting on paper.	Use a keyboard to type letters, words and sentences. Use spacebar, Enter/Return, Backspace and Shift appropriately. Type and edit text using a word processor. Use capital letters and punctuation. Select, delete and replace text. Use copy and paste. Change font, size and style. Use bold, italic and underline.	Change text alignment. Use headings and formatting to organise writing. Insert and edit images alongside text. Use digital tools to plan, draft, revise and improve writing. Save, open and organise digital documents. Understand that digital writing can be edited and changed easily. Create documents for different audiences and purposes.	Choose appropriate commands for a purpose. Use block-based programming to create animations. Join command blocks together to make a sequence. Understand and use algorithms to plan a program. Change values within command blocks and understand their effect. Control sprites using programming commands.	Understand that different sprites can have their own instructions. Select and use backgrounds and sprites. Create a simple animated program from a planned algorithm. Sequence instructions in the correct order. Test programs to check that they work as intended. Identify and correct errors/bugs in a program.

Year 2	Knowledge	Information technology in the world beyond school	Information technology in the world beyond school	Using IT to organise and present data	Using IT to organise and present data	Building sequences in programs	Building sequences in programs
	Skills I can… Key Vocabulary: Technology IT Data Organise Present Program Sequence Command Instruction Algorithm Debug	Know what IT means. Identify different IT devices. Know how IT is used in school and at work. Know how IT can make tasks easier, safer and more accurate. Understand that IT devices can work together. Recognise barcodes and QR codes.	Follow basic IT safety rules. Keep passwords and personal information safe. Ask a trusted adult when unsure. Make safe and sensible choices when using technology. Have a healthy balance when using technology.	Collect and record data using tally charts. Count and compare data. Enter data into a computer. Create simple pictograms. Understand how pictures can represent data. Sort objects using different features. Compare data and spot patterns.	Use charts to answer questions. Present data using pictograms and bar charts. Make data clear and easy to understand. Keep personal information private. Understand that not all information should be shared.	Know what a sequence is. Understand that commands make a program do something. Predict what a program will do. Create simple programs using blocks. Use event blocks to start a program. Understand what an algorithm is. Put commands in the correct order. Use sprites in a program.	Make sprites display text. Use multiple sprites. Change backgrounds. Use tap events to trigger actions. Create a simple quiz. Test and improve a program. Understand that changing the order of commands changes the outcome.

Year 3	Knowledge	Computer networks	Stop-frame animation	Programming sequences using sound	Organising data using databases	Desktop publishing	Events and actions in programs
	Skills I can… Key Vocabulary: Network Internet Device Animation Frame Sequence Program Sound Data Database Table Layout Font Event Action Input Output Debug	Know what a digital device is. Understand input, process and output. Design a simple digital device. Know what a computer network is. Know how devices connect to a network. Understand how information is shared. Understand how data moves between devices. Know what packets are. Identify parts of a network. Know what a switch, server, router and wireless access point do. Use basic network vocabulary.	Know what animation is. Understand what a frame is. Create a simple stop-frame animation. Make small changes between frames. Plan an animation using a storyboard. Keep characters and backgrounds consistent. Review and improve an animation. Add text, pictures and sounds to an animation. Use feedback to improve your work. Use animation software safely and carefully.	Create a simple sequence of instructions. Understand what a sequence is. Use commands to control a program. Add sounds to a program. Change the order of commands. Create a simple music sequence. Find and fix errors in a program. Test and improve a program. Use repetition in a program. Use programming vocabulary.	Know what data is. Understand what a database is. Organise data into groups. Add information to a database. Use fields to organise data. Search for information in a database. Sort and filter data. Use data to answer questions. Present data clearly. Check data for accuracy.	Know what desktop publishing is. Create a simple document. Add and edit text. Add and edit images. Change font, size and colour. Arrange text and images on a page. Use text boxes. Choose suitable layouts. Make a document clear and attractive. Review and improve your work.	Know what an event is. Know what an action is. Understand that events can start a program. Create programs using events and actions. Use inputs to trigger actions. Test a program. Find and fix errors. Change and improve a program. Use simple programming vocabulary.

Year 4	Knowledge	The internet	Audio production	Repetition in programs	Data logging	Photo editing	Using repetition in programming to create a game
	Skills I can… Key Vocabulary: Internet Website Audio Recording Repetition Loop Data Logging Photo Editing Programming Game Debugging	Know what the internet is. Understand that the internet connects networks around the world. Know the difference between the internet and the World Wide Web. Understand how data travels across networks. Know what a router does. Know how websites and information are shared online. Understand that people can create and share content online. Know that online content can be owned by someone. Understand copyright. Check if information online is reliable and trustworthy. Recognise that online information can be false or misleading. Use the internet safely and responsibly.	Know what audio is. Record sound using a computer. Use a microphone to record audio. Edit and improve audio recordings. Cut, move and delete parts of a recording. Add background music and sound effects. Change the volume of sounds. Arrange sounds in the correct order. Create a simple podcast or audio project. Listen to and review your work. Make improvements to an audio project.	Know what repetition means in programming. Use loops to repeat instructions. Create programs using simple loops. Use a count to control repetition. Predict what a repeated program will do. Find and fix errors in a program. Test programs to check they work. Change and improve a program. Understand that loops can make programs shorter and easier. Use simple programming vocabulary.	Know what data logging is. Know what a data logger does. Collect data using a data logger. Use sensors to collect data. Record data automatically. Understand that data can be collected over time. Choose suitable sensors for different tasks. Read and interpret collected data. Present data using tables and graphs. Look for patterns and changes in data. Use data to answer questions. Review and improve how data is collected.	Know what photo editing is. Take or import a digital photo. Crop and resize images. Rotate and flip images. Change brightness and colour. Add effects and filters. Remove or change parts of an image. Add text to a photo. Combine images to create a new picture. Understand that photos can be edited to change their appearance. Review and improve an edited photo. Use images responsibly and respect copyright.	Know what repetition means in programming. Use loops to repeat instructions. Create a game using repetition. Use loops to control game actions. Use repetition to move sprites or characters. Use events to start actions in a game. Predict what repeated instructions will do. Test a game to check it works. Find and fix errors in a program. Change and improve a game. Understand that loops make programs shorter and easier. Use simple programming vocabulary.

Year 5	Knowledge	Introduction to computer systems	Video production	Exploring selection in physical computing	Flat-file databases	Introduction to vector graphics	Using selection in programming to develop a quiz
	Skills I can… Key Vocabulary: Computer Systems Video Physical Computing Selection Database Data Vector Graphics Shapes Programming Quiz Input Output Debugging	Know what a computer system is. Know the main parts of a computer system. Understand the difference between hardware and software. Know what an input is. Know what an output is. Understand how inputs and outputs work together. Identify common input devices. Identify common output devices. Know what the CPU does. Understand how computers process information. Know that computers follow instructions to complete tasks.	Know what video production is. Plan a video using a storyboard. Record video using a camera or device. Record clear video and sound. Import video clips into editing software. Trim and arrange video clips. Add text and titles to a video. Add music and sound effects. Use transitions between clips. Review and improve a video. Export and share a finished video safely. Use images, music and videos responsibly and respect copyright.	Know what selection means in programming. Understand that programs can make decisions. Use if statements in programs. Use if/else statements to make choices. Use inputs to control a physical device. Use sensors to detect changes. Make an output happen when a condition is met. Predict what a program will do. Test a physical computing program. Find and fix errors in a program. Change and improve a program.	Know what a database is. Know what a flat-file database is. Understand what fields and records are. Add and edit information in a database. Search for information in a database. Sort and filter data . Use different fields to find information. Ask questions about the data. Use data to answer questions. Present information clearly. Check data for accuracy. Understand that data should be kept safe and private.	Know what vector graphics are. Understand the difference between vector and bitmap images. Create simple vector graphics. Use shapes to create images. Resize and move shapes. Change the colour and outline of shapes. Layer and group objects. Duplicate and edit shapes. Use tools to combine shapes. Add text to a vector graphic. Arrange objects to create a finished image.	Know what selection means in programming. Understand how programs make decisions. Use if statements in programs. Use if/else statements to make choices. Create a quiz using selection. Use conditions to check answers. Give different responses for correct and incorrect answers. Use variables to store information such as scores. Test a quiz to check it works. Find and fix errors in a program.

Year 6	Knowledge	Communication and the internet	Web page creation	Using variables in programming to develop a game	Introduction to spreadsheets	3D modelling	Sensing movement with physical computing
	Skills I can… Key Vocabulary: Communication Internet Websites Web Pages Programming Variables Games Spreadsheets Data Formulas 3D Modelling Shapes Physical Computing Sensors Movement Input Output Debugging	Know what an IP address is. Understand how internet addresses help devices communicate. Know what DNS does. Understand how data is split into packets. Know that packets are sent across networks and reassembled. Understand how the internet allows people to work together. Use shared files and cloud storage. Understand the difference between public and private online spaces. Choose suitable communication methods for different purposes. Use email, instant messaging, video calls and other online tools appropriately. Communicate safely and responsibly online.	Know what a website and web page are. Understand how web pages are structured. Plan a website for a specific purpose and audience. Create web pages using text and images. Use headings, paragraphs and other page elements. Add links between web pages. Choose suitable images and content. Understand what HTML is used for. Make web pages clear and easy to navigate. Preview and test web pages. Improve a website using feedback. Understand copyright and use content responsibly. Consider accessibility	Know what a variable is. Understand that variables store information. Create and use variables in a program. Change the value of a variable. Use variables to keep score in a game. Use variables to control game features. Use selection and conditions in a game. Use repetition to control game actions. Predict what a program will do. Test and debug a game. Find and fix errors in a program. Improve a game by adding new features. Use programming vocabulary accurately.	Know what a spreadsheet is. Know what rows, columns and cells are. Enter and edit data in a spreadsheet. Use labels and headings to organise data. Use simple formulas to calculate values. Use functions such as SUM and AVERAGE. Understand that formulas can use cell references. Copy and fill formulas into other cells. Format cells to make data clear. Create charts and graphs from data. Use spreadsheets to answer questions. Check data and formulas for accuracy. Review and improve a spreadsheet.	Know what 3D modelling is. Understand the difference between 2D and 3D shapes. Create simple 3D models. Add and edit 3D shapes. Move, resize and rotate objects. Combine shapes to make a model. Change the colour and appearance of objects. Use layers or groups to organise a model. View a model from different angles. Plan a 3D model before creating it. Review and improve a model. Understand how 3D models can be used in real life. Use 3D modelling software safely and responsibly.	Know what physical computing is. Understand how sensors detect movement. Use a motion or light sensor to collect information. Use inputs to control physical devices. Use variables to store sensor readings. Use selection to make decisions based on sensor data. Use repetition to control actions. Create a program that responds to movement. Predict what a physical computing program will do. Test and debug a program. Find and fix errors in a program. Improve a physical computing project.

		Understand that online communication may not be private. Know not to share personal information without permission. Understand the importance of consent and privacy.	when creating web pages.				Use programming and physical computing vocabulary.

Year 7	Knowledge	Computer networks and data transmission	Using fundamental programming constructs in a block based language	Physical computing using the micro: bit	Data modelling	Clear messaging in digital media	Using media to gain support for a cause
	Skills I can… Key Vocabulary: Networks Data Transmission Programming Blocks Algorithms Variables Selection Repetition Micro:bit Sensors Inputs Outputs Data Modelling Graphs Digital Media Messages Communication Audience Campaigns Persuasion	Know what a computer network is. Understand the difference between a LAN and a WAN. Know what the internet is and how networks connect. Understand the role of a router and switch. Know what an IP address is. Understand how devices are identified on a network. Know what packets are. Understand how data is split into packets for transmission. Understand how packets travel across networks. Know that packets can take different routes to their destination. Understand how packets are reassembled at the	Know what a programming language is. Understand what block-based programming is. Know what a sequence is. Use variables to store data. Use inputs and outputs in programs. Use selection to make decisions. Use if and if/else statements. Use repetition and loops. Use for and while loops where appropriate. Create programs using different programming constructs. Predict what a program will do. Test programs using different inputs.	Know what physical computing is. Know what a micro:bit is and how it can be used. Identify the inputs and outputs on a micro:bit. Use buttons and sensors as inputs. Use the LED display as an output. Create programs for the micro:bit. Use variables to store information. Use selection to make decisions. Use repetition and loops. Use sensor data to control outputs. Create programs that respond to user input. Test and debug micro:bit programs. Find and fix errors in	Know what data modelling is. Understand how models can represent real-world situations. Identify inputs, outputs and variables in a model. Enter and organise data in a spreadsheet. Use formulas to calculate values. Use cell references in formulas. Use functions such as SUM, AVERAGE and IF. Change input values to explore different outcomes. Use models to make predictions. Create charts and graphs to present results. Identify patterns and trends in data. Check whether a model gives sensible results.	Know what digital media is. Understand how digital media can communicate a message. Identify the purpose and audience of a digital message. Choose suitable text, images, audio and video. Use clear and appropriate language. Organise information so it is easy to understand. Use headings, layout and design effectively. Understand how colour, fonts and images affect a message. Make digital content accessible to different audiences. Consider how different platforms affect communication.	Know what a cause is. Understand how digital media can raise awareness. Identify the target audience for a campaign. Research information about a cause. Choose suitable media for a campaign. Create persuasive digital content. Use text, images, audio and video effectively. Use facts and evidence to support a message. Understand how language and images can influence people. Create a clear call to action. Consider different viewpoints and perspectives. Check information is

		destination. Know what protocols are and why they are needed. Understand the purpose of TCP/IP. Understand how data can be transmitted using wired and wireless connections. Explain how network performance can be affected by factors such as bandwidth and traffic. Use appropriate network and data-transmission vocabulary.	Find and fix errors using debugging. Break a problem into smaller parts. Use algorithms to solve problems. Use programming vocabulary accurately.	code. Improve a physical computing project. Use programming and physical computing vocabulary accurately.	Understand that models are simplified versions of real situations. Evaluate the strengths and limitations of a model. Use data modelling vocabulary accurately.	Review and improve digital media using feedback. Understand copyright and use digital content responsibly. Communicate safely and respectfully online.	accurate and reliable. Understand copyright and use media responsibly. Review and improve a campaign using feedback. Communicate respectfully and responsibly online.

Year 8	Knowledge	Computer systems and data science	Data representation: text and numbers	Developing vector graphics	Developing for the Web	Mobile app development	Introduction to Python programming
	Skills I can… Key Vocabulary: Networks Devices Data Transmission Connections Internet Packets Protocols Servers Routers Security Communication	Know what a computer system is. Know the main parts of a computer. Understand how hardware and software work together. Know what the CPU, memory and storage do. Understand how computers process data. Know what data science is. Collect and organise data. Analyse data to find patterns. Use charts and graphs to show data. Use data to make predictions. Check if data is accurate and reliable. Understand that data can be biased. Use simple data science vocabulary.	Know that computers store data as binary. Understand what bits and bytes are. Know how text is represented using character codes. Understand that different characters have different binary codes. Know how numbers are represented in binary. Convert simple numbers between binary and denary. Understand how much data can be stored using bits and bytes. Know the difference between text and numerical data. Understand that computers can only store data in a limited number of bits. Identify errors in data representation. Use simple binary and data representation vocabulary.	Know what vector graphics are. Understand how vector images are made from shapes and objects. Create and edit vector graphics. Use shapes, lines and curves to create images. Resize, rotate and move objects. Change colours, fills and outlines. Layer and group objects. Combine and edit shapes to create more complex designs. Use text in vector graphics. Create designs for a specific purpose and audience. Export vector graphics in suitable file formats. Review and improve a design. Use images and graphics responsibly.	Know what a website and web page are. Understand how websites are structured. Know what HTML is used for. Create simple web pages using HTML. Add headings, paragraphs and images. Add links between web pages. Use basic HTML tags correctly. Plan a website for a specific audience. Organise content clearly on a web page. Test web pages to check they work. Improve a website using feedback. Understand copyright and use content responsibly. Consider accessibility when creating web pages.	Know what a mobile app is. Understand how mobile apps are designed and developed. Plan an app for a specific purpose and audience. Design simple app screens. Create an app using a development tool. Add buttons, text and images. Use inputs and outputs in an app. Use variables to store information. Use selection to make decisions. Use events to control what happens in an app. Test an app and find errors. Improve an app using feedback. Use apps safely and responsibly.	Know what Python is. Understand what a programming language is. Write simple Python programs. Use variables to store data. Use different types of data. Get input from the user. Display output using print. Use calculations and basic operators. Use if statements to make decisions. Use loops to repeat instructions. Test and debug Python programs. Find and fix simple errors. Use basic Python programming vocabulary.

Year 9	Knowledge	Data representation: images and sound	Using data science	Machine learning using the micro:bit	Python programming with sequences of data	Introduction to cybersecurity	3D animation
	Skills I can… Key Vocabulary: Images Sound Data Data Science Machine Learning Micro:bit Sensors Python Sequences Lists Cybersecurity Passwords Online Safety 3D Animation Models Movement Programming Algorithms Debugging	Know that digital images are made from pixels. Understand how images are stored using binary. Know how colours are represented using RGB. Understand how image resolution affects quality and file size. Know how sound is stored using binary. Understand what sample rate means. Understand what bit depth means. Know how sample rate and bit depth affect sound quality. Understand what compression is. Know the difference between lossy and lossless compression.	Know what data science is. Understand how data can be collected. Collect and organise data. Clean data to remove errors. Use spreadsheets or other tools to analyse data. Find patterns and trends in data. Use charts and graphs to present data. Use data to answer questions. Use data to make predictions. Understand that data can be biased. Check if data is accurate and reliable. Understand how data can be used to support decisions.	now what machine learning is. Understand how computers learn from data. Know what training data is. Collect data using a micro:bit. Use sensors to collect data. Label and organise training data. Train a simple machine learning model. Use a model to make predictions. Test how well a model works. Understand that poor data can lead to poor results. Understand that data can be biased. Use a machine learning model with a	Know what a sequence of data is. Use lists to store multiple values. Create and edit lists in Python. Access individual items in a list. Add and remove items from a list. Use loops to work through a list. Use variables to store data. Use input and output with lists. Search and process data in a list. Use conditions with lists. Test and debug Python programs. Use sequences of data to solve problems. Use basic Python programming	Know what cybersecurity is. Understand why cybersecurity is important. Know common types of cyber threats. Understand what malware is. Know how phishing attacks work. Understand how strong passwords help protect accounts. Know what multi-factor authentication is. Understand how encryption protects data. Know how firewalls help protect networks. Understand how social engineering can trick people. Know how to keep personal data safe. Recognise common signs of a cyber attack. Know how to respond	Know what 3D animation is. Understand how 3D models are used in animation. Create simple 3D models. Add and edit shapes and objects. Move, rotate and resize objects. Create and position a camera. Add lighting to a scene. Create simple animations using keyframes. Change the position and movement of objects over time. Add materials, colours and textures. Preview and review an animation. Improve an animation using

		Understand how compression can reduce file size. Use basic image and sound data vocabulary.	Use basic data science vocabulary.	micro:bit. Use basic machine learning vocabulary.	vocabulary.	to a security threat. Use basic cybersecurity vocabulary.	feedback. Export a finished 3D animation.

Year 10	Knowledge	Computer Systems	Programming: sequence	Programming: selection	Programming: iteration	Programming: subprograms	Programming: strings and lists
	Skills I can… Key Vocabulary: Computer Systems Programming Sequence Selection Iteration Loops Subprograms Functions Strings Lists Variables Algorithms Inputs Outputs Debugging	Know what a computer system is. Know the difference between hardware and software. Know what an operating system does. Know the main parts of a CPU. Understand how the CPU processes instructions. Know what RAM, ROM and cache do. Know different types of storage. Understand how storage devices are used. Know how CPU speed affects performance. Understand what embedded systems are. Know what e-waste is. Understand the environmental impact of computers. Use basic computer systems vocabulary.	Know what a sequence is in programming. Understand that instructions run in order. Write simple programs using sequence. Use variables to store data. Use different data types. Use input and output. Use arithmetic operators. Use assignment statements. Trace a program step by step. Predict the output of a program. Test programs using different inputs. Find and fix simple errors. Use basic programming vocabulary.	Know what selection means in programming. Understand how programs make decisions. Use if statements. Use if...else statements. Use comparison operators. Use logical operators such as AND, OR and NOT. Use conditions to control program flow. Combine selection with variables and input. Predict what a program will do. Test programs with different inputs. Find and fix errors . Use basic programming vocabulary.	Know what iteration means in programming. Understand why loops are used. Use for loops. Use while loops. Use conditions to control loops. Use loops with variables. Use loops with lists or strings. Predict what a loop will do. Trace a program step by step. Test programs with different inputs.	Know what a subprogram is. Understand why subprograms are used. Create and use functions. Create and use procedures. Pass data into a subprogram using parameters. Return values from functions. Use subprograms to break a problem into smaller parts. Reuse code using subprograms. Understand local and global variables. Test subprograms with different inputs. Find and fix errors in subprograms. Use basic programming vocabulary.	Know what a string is. Know what a list is. Create and use strings and lists. Access individual characters and items. Add, remove and change items in a list . Use string operations to work with text. Use loops with strings and lists. Search for items in a list. Sort and organise list data. Use lists to store multiple values. Test programs using strings and lists. Find and fix errors. Use basic programming vocabulary.

Year 11	Knowledge	Programming: dictionaries and data files	Python programming project	HTML programming project	Databases and SQL		
	Skills I can… Key Vocabulary: Dictionary Key Value CSV File Algorithm Variable Selection Iteration Function HTML Tag Element Attribute Hyperlink Database Table Field SQL Query	Know what a dictionary is. Understand key-value pairs. Create and use dictionaries in Python. Use dictionaries to store records. Access and change dictionary values. Use dictionaries in programs. Read data from text files. Write data to text files. Understand read, write and append modes. Read and write CSV files. Use file data in a program. Test and debug programs using dictionaries and files. Use basic programming and data vocabulary.	Plan a Python program to solve a problem. Break a problem into smaller parts. Use variables and different data types. Use sequence, selection and iteration. Use strings and lists. Use dictionaries and data files. Create and use subprograms. Use input and output. Validate user input. Test programs with different inputs. Find and fix errors using debugging. Improve a program based on testing and feedback. Explain how the program works. Use clear and organised Python code.	Plan a website for a specific purpose and audience. Create web pages using HTML. Use headings, paragraphs and lists. Add images and other media. Create links between pages. Use tables to organise information. Use forms to collect user input. Use suitable HTML tags. Structure web pages clearly. Test links and page content. Find and fix errors in HTML. Improve pages using feedback. Consider accessibility and usability. Use copyright-friendly content. Explain and evaluate the finished website.	Know what a database is. Understand tables, records and fields. Identify primary keys. Create and edit database tables. Add, change and delete records. Use SQL to search for data. Use SELECT to retrieve data. Use WHERE to filter data. Use ORDER BY to sort data. Use AND, OR and NOT in queries. Use calculations and functions in SQL. Add, update and delete records using SQL. Test SQL queries and check results. Use databases to store and manage data.		