

St Dominic's School

Curriculum Plan

What will pupils learn and when?



Subject	KS3 Computing	Group	Year 8
Subject Intent	The KS3 curriculum provides challenges and new experiences in computing, digital literacy and digital media, regardless of pupils' prior knowledge of using computers, and is designed to ensure that those going on to study GCSE Computer Science have a basis of knowledge, skills and understanding in the fundamental concepts covered at KS4. Over time, students learn to program in 3 languages, starting with block-based languages before progressing to High-Level Languages. The development of programming skills is also built into physical Computing tasks using the BBC Micro:Bits, for example coding LED lights to effectively apply the knowledge learnt in earlier Algorithm and Programming units. The curriculum journey connects to other curriculum areas holistically to ensure learning contexts are authentic, meaningful and provide opportunities for application of skills, investigation and purposeful play. In addition, references to key events and developments through the history of technology using role models from all aspects of society are used in an inspirational and motivational way for students. The scheme of work for Computing is adapted from the 'Teach Computing' Curriculum and covers all aspects of the National Curriculum. This scheme was chosen as it has been created by subject experts and based on the latest pedagogical research. It provides an innovative progression framework where computing content (concepts, knowledge, skills and objectives) has been organised into interconnected networks called learning graphs. The curriculum aims to equip young people with the knowledge, skills and understanding they need to thrive in the digital world of today and the future. The national curriculum for computing aims to ensure all pupils: • 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 2 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 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] • 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) can 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, reuse, revise and repurpose 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		

St Dominic's School

Curriculum Plan

What will pupils learn and when?



Term	Topic	Subject Content	Vocabulary	Sequencing
Autumn 1	Computer Systems and Networks: Computing Systems	⌘ I can recall that a general-purpose computing system is a device for executing programs ⌘ I can recall that a program is a sequence of instructions that specify operations that are to be performed on data ⌘ I can explain the difference between a general-purpose computing system and a purpose-built device ⌘ I can describe the function of the hardware components used in computing systems ⌘ I can describe how the hardware components used in computing systems work together in order to execute programs ⌘ I can recall that all computing systems, regardless of form, have a similar structure ('architecture') ⌘ I can analyse how the hardware components used in computing systems work together in order to execute programs ⌘ I can define what an operating system is, and recall its role in controlling program execution ⌘ I can describe the NOT, AND, and OR logical operators, and how they are used to form logical expressions	Tier 2: System Program Operations Hardware Components Artificial intelligence Tier 3: General purpose Operating system Logic gates Logic circuits Binary digits Machine learning	This unit takes pupils on a tour through the different layers of computing systems: from programs and the operating system, to the physical components that store and execute these programs, to the fundamental binary building blocks that these components consist of. The aim is to provide a concise overview of how computing systems operate, conveying the essentials and abstracting away the technical details that might confuse or put off pupils. The last lessons cover two interesting contemporary topics: artificial intelligence and open source software. These are linked back to the content of the unit, helping pupils to both broaden their knowledge and focus on the topics addressed in the unit. The unit assumes no prior knowledge. There are, however, links to the 'Representations' units taught in Years 8 and 9 and the 'Networks' units taught in Years 7 and 8.

St Dominic's School

Curriculum Plan

What will pupils learn and when?



		✗ I can use logic gates to construct logic circuits, and associate these with logical operators and expressions ✗ I can describe how hardware is built out of increasingly complex logic circuits ✗ I can recall that, since hardware is built out of logic circuits, data and instructions alike need to be represented using binary digits ✗ I can provide broad definitions of 'artificial intelligence' and 'machine learning' ✗ I can identify examples of artificial intelligence and machine learning in the real world ✗ I can describe the steps involved in training machines to perform tasks (gathering data, training, testing) ✗ I can describe how machine learning differs from traditional programming ✗ I can associate the use of artificial intelligence with moral dilemmas ✗ I can explain the implications of sharing program code		
Autumn 2	Computer Systems and Networks Developing for the Web	✗ I can describe what HTML is ✗ I can use HTML to structure static web pages ✗ I can modify HTML tags using inline styling to improve the appearance of web pages ✗ I can display images within a web page ✗ I can apply HTML tags to construct a web page structure from a provided design		In this unit, pupils will explore the technologies that make up the internet and World Wide Web. Starting with an exploration of the building blocks of the World Wide Web, HTML, and CSS, pupils will investigate how websites are catalogued and organised for effective retrieval using search engines. By

St Dominic's School

Curriculum Plan

What will pupils learn and when?



		✎ I can describe what CSS is ✎ I can use CSS to style static web pages ✎ I can assess the benefits of using CSS to style pages instead of in-line formatting ✎ I can describe what a search engine is ✎ I can explain how search engines 'crawl' through the World Wide Web and how they select and rank results ✎ I can analyse how search engines select and rank results when searches are made ✎ I can use search technologies effectively ✎ I can discuss the impact of search technologies and the issues that arise by the way they function and the way they are used ✎ I can create hyperlinks to allow users to navigate between multiple web pages ✎ I can implement navigation to complete a functioning website		the end of the unit, pupils will have a functioning website.
Spring 1	Media: Vector Graphics	✎ I can draw basic shapes (rectangle, ellipse, polygon, star) with different properties (fill and stroke, shape-specific attributes) ✎ I can manipulate individual objects (select, move, resize, rotate, duplicate, flip, z-order) ✎ I can manipulate groups of objects (select, group/ungroup, align, distribute) ✎ I can combine paths by applying operations (union, difference, intersection) ✎ I can convert objects to paths		This unit offers students the opportunity to design graphics using vector graphic editing software. Vector graphics can be used to design anything from logos and icons to posters, board games, and complex illustrations. Through this unit, students will be able to better understand the processes involved in creating such graphics and will be provided with the knowledge and tools to create their own.

St Dominic's School

Curriculum Plan

What will pupils learn and when?



		✎ I can draw paths ✎ I can edit path nodes ✎ I can combine multiple tools and techniques to create a vector graphic design ✎ I can explain what vector graphics are ✎ I can provide examples where using vector graphics would be appropriate ✎ I can peer assess another pair's project work ✎ I can improve your own project work based on feedback		One of the most interesting and challenging aspects of creating vector graphics is their unlikely link to computational thinking. Creating a complex design is a multi-step process that starts with elementary shapes and involves combining them into more intricate ones using operations such as union, difference, and intersection. There are usually multiple paths to achieving the goal and the process involves decomposition, evaluation, and plenty of inventiveness!
Spring 2	Data and Information: Representations, from clay to silicon	✎ I can list examples of representations ✎ I can recall that representations are used to store, communicate, and process information ✎ I can provide examples of how different representations are appropriate for different tasks ✎ I can recall that characters can be represented as sequences of symbols and list examples of character coding schemes ✎ I can measure the length of a representation as the number of symbols that it contains ✎ I can provide examples of how symbols are carried on physical media ✎ I can explain what binary digits (bits) are, in terms of familiar symbols such as digits or letters		This unit conveys essential knowledge relating to binary representations. The activities gradually introduce learners to binary digits and how they can be used to represent text and numbers. The concepts are linked to practical applications and problems that the learners are familiar with.

St Dominic's School

Curriculum Plan

What will pupils learn and when?



		⌘ I can measure the size or length of a sequence of bits as the number of binary digits that it contains ⌘ I can describe how natural numbers are represented as sequences of binary digits ⌘ I can convert a decimal number to binary and vice versa ⌘ I can convert between different units and multiples of representation size ⌘ I can provide examples of the different ways that binary digits are physically represented in digital devices		
Summer 1	Programming: Introduction to Python programming	⌘ I can describe what algorithms and programs are and how they differ ⌘ I can recall that a program written in a programming language needs to be translated in order to be executed by a machine ⌘ I can write simple Python programs that display messages, assign values to variables, and receive keyboard input ⌘ I can locate and correct common syntax errors ⌘ I can describe the semantics of assignment statements ⌘ I can use simple arithmetic expressions in assignment statements to calculate values ⌘ I can receive input from the keyboard and convert it to a numerical value		This unit introduces learners to text-based programming with Python. The lessons form a journey that starts with simple programs involving input and output, and gradually moves on through arithmetic operations, randomness, selection, and iteration. Emphasis is placed on tackling common misconceptions and elucidating the mechanics of program execution. A range of pedagogical tools is employed throughout the unit, with the most prominent being pair programming, live coding, and worked examples. The Year 7 Programming units (Scratch Essentials I and Scratch Essentials II) are a prerequisite for this unit.

St Dominic's School

Curriculum Plan

What will pupils learn and when?



		✗ I can use relational operators to form logical expressions ✗ I can use binary selection (if, else statements) to control the flow of program execution ✗ I can generate and use random integers ✗ I can use multi-branch selection (if, elif, else statements) to control the flow of program execution ✗ I can describe how iteration (while statements) controls the flow of program execution ✗ I can use iteration (while loops) to control the flow of program execution ✗ I can use variables as counters in iterative programs ✗ I can combine iteration and selection to control the flow of program execution ✗ I can use Boolean variables as flags		
Summer 2	Programming: Mobile App development	✗ I can identify when a problem needs to be broken down ✗ I can implement and customise GUI elements to meet the needs of the user ✗ I can recognise that events can control the flow of a program ✗ I can use user input in an event-driven programming environment ✗ I can use variables in an event-driven programming environment ✗ I can develop a partially complete application to include additional functionality		In a world where there's an app for every possible need, this unit aims to take the learners from designer to project manager to developer in order to create their own mobile app. Using App Lab from code.org, learners will familiarise themselves with the coding environment and have an opportunity to build on the programming concepts they used in previous units before undertaking their project. Learners will work in pairs to consider the needs of the user; decompose the project into smaller, more manageable parts; use the

St Dominic's School

Curriculum Plan

What will pupils learn and when?



		€ I can identify and fix common coding errors € I can pass the value of a variable into an object € I can establish user needs when completing a creative project € I can apply decomposition to break down a large problem into more manageable steps € I can use user input in a block-based programming language € I can use a block-based programming language to create a sequence € I can use variables in a block-based programming language € I can use a block-based programming language to include sequencing and selection € I can use user input in a block-based programming language € I can use variables in a block-based programming language € I can reflect and react to user feedback € I can use a block-based programming language to include sequencing and selection € I can use user input in a block-based programming language € I can use variables in a block-based programming language € I can evaluate the success of the programming project		pair programming approach to develop their app together; and finish off by evaluating the success of the project against the needs of the user.
--	--	--	--	--

St Dominic's School

Curriculum Plan

What will pupils learn and when?

