

St Dominic's School

Curriculum Plan

What will pupils learn and when?



Subject	KS3 Computing	Group	Year 9
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: Cybersecurity	≠ I can explain the difference between data and information ≠ I can critique online services in relation to data privacy ≠ I can identify what happens to data entered online ≠ I can explain the need for the Data Protection Act ≠ I can recognise how human errors pose security risks to data ≠ I can implement strategies to minimise the risk of data being compromised through human error ≠ I can define hacking in the context of cyber security ≠ I can explain how a DDoS attack can impact users of online services ≠ I can identify strategies to reduce the chance of a brute force attack being successful ≠ Explain the need for the Computer Misuse Act ≠ I can list the common malware threats ≠ I can examine how different types of malware causes problems for computer systems ≠ I can question how malicious bots can have an impact on societal issues ≠ I can compare security threats against probability and the potential impact to organisations	Tier 2: Data Information Privacy Virus Tier 3: Data protection act Computer misuse act DDoS Brute force Social engineering Blagging Shouldering Phishing Pharming Malware	This unit takes the pupils on an eye-opening journey of discovery about techniques used by cybercriminals to steal data, disrupt systems, and infiltrate networks. The pupils will start by considering the value of their data to organisations and what they might use it for. They will then look at social engineering techniques used by cybercriminals to try to trick users into giving away their personal data. The unit will look at the more common cybercrimes such as hacking, DDoS attacks, and malware, as well as looking at methods to protect ourselves and our networks against these attacks.

St Dominic's School

Curriculum Plan

What will pupils learn and when?



		€ I can explain how networks can be protected from common security threats € I can identify the most effective methods to prevent cyberattacks		
Autumn 2	Computer Systems and Networks Data Science	€ I can define data science € I can explain how visualising data can help identify patterns and trends in order to help us gain insights € I can use an appropriate software tool to visualise data sets and look for patterns or trends € I can recognise examples of where large data sets are used in daily life € I can select criteria and use data set to investigate predictions € I can evaluate findings to support arguments for or against a prediction € I can define the terms 'correlation' and 'outliers' in relation to data trends € I can identify the steps of the investigative cycle € I can solve a problem by implementing steps of the investigative cycle on a data set € I can use findings to support a recommendation € I can identify the steps of the investigative cycle € I can identify the data needed to answer a question defined by the pupil € I can create a data capture form € I can describe the need for data		In this unit, pupils will be introduced to data science, and by the end of the unit they will be empowered by knowing how to use data to investigate problems and make changes to the world around them. Pupils will be exposed to both global and local data sets and gain an understanding of how visualising data can help with the process of identifying patterns and trends. Towards the end of the unit, the pupils will go through the steps of the investigative cycle to try to solve a problem in the school using data.

St Dominic's School

Curriculum Plan

What will pupils learn and when?



		cleansing € I can apply data cleansing techniques to a data set € I can visualise a data set € I can analyse visualisations to identify patterns, trends, and outliers € I can draw conclusions and report findings		
Spring 1	Media: Animations	€ I can add, delete, and move objects € I can scale and rotate objects € I can use a material to add colour to objects € I can add, move, and delete keyframes to make basic animations € I can play, pause, and move through the animation using the timeline € I can create useful names for objects € I can join multiple objects together using parenting € I can use edit mode and extrude € I can use loop cut and face editing € I can apply different colours to different parts of the same model € I can use proportional editing € I can use the knife tool € I can use subdivision € I can add and edit set lighting € I can set up the camera € I can compare different render modes € I can create a 3–10 second animation € I can render out an animation		Films, television, computer games, advertising, and architecture have been revolutionised by computer-based 3D modelling and animation. In this unit learners will discover how professionals create 3D animations using the industry-standard software package, Blender. By completing this unit learners will gain a greater understanding of how this important creative field is used to make the media products that we consume. Sessions will take learners through the basics of modelling, texturing, and animating; outputs will include 3D models, short videos, and VR. Links are made throughout to computer science, computational thinking, and the world of work. Tools and techniques learnt in this unit can also be used for 3D printing.

St Dominic's School

Curriculum Plan

What will pupils learn and when?



Spring 2	Data and Representations, going audiovisual	€ I can describe how digital images are composed of individual elements € I can recall that the colour of each picture element is represented using a sequence of binary digits € I can define key terms such as 'pixels', 'resolution', and 'colour depth' € I can describe how an image can be represented as a sequence of bits € I can describe how colour can be represented as a mixture of red, green, and blue, with a sequence of bits representing each colour's intensity € I can compute the representation size of a digital image, by multiplying resolution (number of pixels) with colour depth (number of bits used to represent the colour of individual pixels) € I can describe the trade-off between representation size and perceived quality for digital images € I can perform basic image editing tasks using appropriate software and combine them in order to solve more complex problems requiring image manipulation € I can explain how the manipulation of digital images amounts to arithmetic operations on their digital representation € I can describe and assess the creative benefits and ethical drawbacks of digital manipulation (Education for a Connected World) € I can recall that sound is a wave		In this unit, learners will focus on digital media such as images and sounds, and discover the binary digits that lie beneath these types of media. Just like in the previous unit, where learners examined characters and numbers, the ideas that learners need to understand are not really new to them. You will draw on familiar examples of composing images out of individual elements, mixing elementary colours to produce new ones, and taking samples of analogue signals, to illustrate these ideas and bring them together in a coherent narrative. This unit also has a significant practical aspect. Learners will use relevant software (GIMP and Audacity, in this case) to manipulate images and sounds and get an idea of how the underlying principles of digital representations are applied in real settings. This unit builds on the material from the Year 8 unit, 'Representations: from clay to silicon'.
----------	---	---	--	--

St Dominic's School

Curriculum Plan

What will pupils learn and when?



		€ I can explain the function of microphones and speakers as components that capture and generate sound € I can define key terms such as 'sample', 'sampling frequency/rate', 'sample size' € I can describe how sounds are represented as sequences of bits € I can calculate representation size for a given digital sound, given its attributes € I can explain how attributes such as sampling frequency and sample size affect characteristics such as representation size and perceived quality, and the trade-offs involved € I can perform basic sound editing tasks using appropriate software and combine them in order to solve more complex problems requiring sound manipulation € I can recall that bitmap images and pulse code sound are not the only binary representations of images and sound available € I can define 'compression', and describe why it is necessary		
Summer 1	Programming: Python programming with sequences of data	€ I can write programs that display messages, receive keyboard input, and use simple arithmetic expressions in assignment statements € I can use selection (if-elif-else statements) to control the flow of program execution		This unit introduces pupils to how data can be represented and processed in sequences, such as lists and strings. The lessons cover a spectrum of operations on sequences of data, that range from accessing an individual element to manipulating the entire sequence. Great care has been taken so that the

St Dominic's School

Curriculum Plan

What will pupils learn and when?



		€ I can locate and correct common syntax errors € I can create lists and access individual list items € I can perform common operations on lists or individual items € I can use iteration (while statements) to control the flow of program execution € I can perform common operations on lists or individual items € I can perform common operations on strings or individual characters € I can use iteration (for statements) to iterate over list items € I can perform common operations on lists or strings € I can use iteration (for loops) to iterate over lists and strings € I can use variables to keep track of counts and sums € I can combine key programming language features to develop solutions to meaningful problems		selection of problems used in the programming tasks are realistic and engaging: pupils will process solar system planets, book texts, capital cities, leaked passwords, word dictionaries, ECG data, and more. A range of pedagogical tools are employed throughout the unit, with the most prominent being pair programming, live coding, and worked examples. The Year 7 and 8 Programming units are prerequisites for this unit. It is assumed that pupils are already able to write Python programs that display messages, receive keyboard input, use simple arithmetic expressions, and control the flow of program execution through selection and iteration structures.
Summer 2	Programming: Physical Computing	€ I can describe what the micro:bit is € I can list the micro:bit's input and output devices € I can use a development environment to write, execute, and debug a Python program for the micro:bit € I can write programs that use the micro:bit's built-in input and output devices		This unit applies and enhances the learners' programming skills in a new engaging context: physical computing, using the BBC micro:bit. In the first half of the unit, learners will get acquainted with the host of components built into the micro:bit, and write simple programs that use these components to interact with

St Dominic's School

Curriculum Plan

What will pupils learn and when?



		€ I can write programs that use GPIO pins to generate output and receive input € I can write programs that communicate with other devices by sending and receiving messages wirelessly € I can design a physical computing artifact purposefully, keeping in mind the problem at hand, the needs of the audience involved, and the available resources € I can decompose the functionality of a physical computing system into simpler features € I can implement a physical computing project, while following, revising, and refining the project plan	the physical world. In the process, they will refresh their Python programming skills and encounter a range of programming patterns that arise frequently in physical computing applications. In the second half, learners will build a physical computing project. They will be required to select and design their project purposefully, apply what they have learnt by building a prototype, and keep a structured diary throughout the process. The Year 8 and 9 programming units are prerequisites for this unit. It is assumed that learners are already able to write Python programs that use variables and data structures to keep track of information. They are also expected to be able to combine sequence, selection, iteration, and function/method calls to control the flow of program execution.
--	--	--	--