**CURRICULUM OVERVIEW 2024 – 2025**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **YR 9** | **Autumn 1** | **Autumn 2** | **Spring 1** | **Spring 2** | **Summer 1** | **Summer 2** |
| **Content** | Python programming with sequences of data | Media – Animations | Data science | Representations – going audiovisual | Cybersecurity | Physical computing (to be changed to An Hour of Code soon) |
| **Key new knowledge** | - Write programs that display messages, receive keyboard input, and use simple arithmetic expressions in assignment statements  - Locate and correct common syntax errors  - Create lists and access individual list items  - Use selection (\*\*if-elif-else\* statements) to control the flow of program execution  - Perform common operations on lists or individual items  - Use iteration (while statements) to control the flow of program execution  - Perform common operations on lists or individual items  - Perform common operations on strings or individual characters  - Use iteration (for statements) to iterate over list items  - Perform common operations on lists or strings  - Use iteration (for loops) to iterate over lists and strings  - Use variables to keep track of counts and sums  - Combine key programming language features to develop solutions to meaningful problems  - Apply all of the skills covered in this unit | - Add, delete, and move objects  - Scale and rotate objects  - Use a material to add colour to objects  - Add, move, and delete keyframes to make basic animations  - Play, pause, and move through the animation using the timeline  - Create useful names for objects  - Join multiple objects together using parenting  - Use edit mode and extrude  - Use loop cut and face editing  - Apply different colours to different parts of the same model  - Use proportional editing  - Use the knife tool  - Use subdivision  - Add and edit set lighting  - Set up the camera  - Compare different render modes  - Create a 3–10 second animation  - Render out the animation | - Define data science  - Explain how visualising data can help identify patterns and trends in order to help us gain insights  - Use an appropriate software tool to visualise data sets and look for patterns or trends  - Recognise examples of where large data sets are used in daily life  - Select criteria and use data set to investigate predictions  - Evaluate findings to support arguments for or against a prediction  - Define the terms ‘correlation’ and ‘outliers’ in relation to data trends  - Identify the steps of the investigative cycle  - Solve a problem by implementing steps of the investigative cycle on a data set  - Use findings to support a recommendation  - Identify the steps of the investigative cycle  - Identify the data needed to answer a question defined by the learner  - Create a data capture form  - Describe the need for data cleansing  - Apply data cleansing techniques to a data set  - Visualise a data set  - Visualise a data set  - Analyse visualisations to identify patterns, trends, and outliers  - Draw conclusions and report findings | - Describe how digital images are composed of individual elements  - Recall that the colour of each picture element is represented using a sequence of binary digits  - Define key terms such as ‘pixels’, ‘resolution’, and ‘colour depth’  - Describe how an image can be represented as a sequence of bits  - Describe how colour can be represented as a mixture of red, green, and blue, with a sequence of bits representing each colour’s intensity  - 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)  - Describe the trade-off between representation size and perceived quality for digital images  - Perform basic image editing tasks using appropriate software and combine them in order to solve more complex problems requiring image manipulation  - Explain how the manipulation of digital images amounts to arithmetic operations on their digital representation  - Describe and assess the creative benefits and ethical drawbacks of digital manipulation [Education for a Connected World](https://www.gov.uk/government/publications/education-for-a-connected-world)  - Recall that sound is a wave  - Explain the function of microphones and speakers as components that capture and generate sound  - Define key terms such as ‘sample’, ‘sampling frequency/rate’, ‘sample size’  - Describe how sounds are represented as sequences of bits  - Calculate representation size for a given digital sound, given its attributes  - Explain how attributes such as sampling frequency and sample size affect characteristics such as representation size and perceived quality, and the trade-offs involved  "- Perform basic sound editing tasks using appropriate software and combine them in order to solve more complex problems requiring sound manipulation  "  - Recall that bitmap images and pulse code sound are not the only binary representations of images and sound available  - Define ‘compression’, and describe why it is necessary | - Explain the difference between data and information  - Critique online services in relation to data privacy  - Identify what happens to data entered online  - Explain the need for the Data Protection Act  - Recognise how human errors pose security risks to data  - Implement strategies to minimise the risk of data being compromised through human error  - Define hacking in the context of cyber security  - Explain how a DDoS attack can impact users of online services  - Identify strategies to reduce the chance of a brute force attack being successful  - Explain the need for the Computer Misuse Act  - List the common malware threats  - Examine how different types of malware causes problems for computer systems  - Question how malicious bots can have an impact on societal issues  - Compare security threats against probability and the potential impact to organisations  - Explain how networks can be protected from common security threats  - Identify the most effective methods to prevent cyberattacks | - Describe what the micro:bit is  - List the micro:bit’s input and output devices  - Use a development environment to write, execute, and debug a Python program for the micro:bit  - Write programs that use the micro:bit’s built-in input and output devices  - Write programs that use GPIO pins to generate output and receive input  - Write programs that communicate with other devices by sending and receiving messages wirelessly  - Design a physical computing artifact purposefully, keeping in mind the problem at hand, the needs of the audience involved, and the available resources  - Decompose the functionality of a physical computing system into simpler features  - Implement a physical computing project, while following, revising, and refining the project plan  - Implement a physical computing project, while following, revising, and refining the project plan |
| **Assessments** | Formative questioning and  teacher observation.  Summative end of unit  assessment. | Formative questioning and  teacher observation.  Summative end of unit  assessment. | Formative questioning and  teacher observation.  Summative end of unit  assessment. | Formative questioning and  teacher observation.  Summative end of unit  assessment. | Formative questioning and  teacher observation.  Summative end of unit  assessment. | Formative questioning and  teacher observation.  Summative end of unit  assessment. |