

## Curriculum Plan KS3 – 2021/2022

Topic	YEAR 9 – CYCLE 1 - Computational Thinking & Logic	
<b>Learning Objectives</b>	<p><b>At the end of this Unit all students should be able to:</b></p> <ul style="list-style-type: none"> <li>• Be able to ask logical questions to solve problems</li> <li>• Know the common Boolean operators:               <ul style="list-style-type: none"> <li>– AND</li> <li>– OR</li> <li>– NOT</li> </ul> </li> <li>• Know different logic gates including:               <ul style="list-style-type: none"> <li>– AND gates</li> <li>– OR gates</li> <li>– NOT gates</li> </ul> </li> <li>• Understand what an algorithm is</li> <li>• Create a sequence of instructions to achieve a goal</li> </ul> <p><b>Most students will be able to:</b></p> <ul style="list-style-type: none"> <li>• Understand how Boolean operators can be represented in written expressions and Venn diagrams</li> <li>• Understand how logic is used in different situations</li> <li>• Be able to complete truth tables for logic gates and circuits with up to three inputs</li> <li>• Understand how loops can be used to reduce the amount of code required for a solution</li> <li>• Be able to refine algorithms to reduce the number of instructions required</li> <li>• Understand the difference between lossy and lossless compression</li> <li>• Be able to use an algorithm to communicate data</li> </ul>	<ul style="list-style-type: none"> <li>– Understand how the algorithm can be improved</li> <li>– Use a binary tree to further improve the algorithm</li> </ul> <ul style="list-style-type: none"> <li>• Understand why compression is needed for video transmission and photo storage</li> <li>• Understand how abstractions are used in everyday life</li> <li>• Be able to create abstractions for different purposes</li> <li>• Understand how networks are used to make an abstraction of a maze</li> <li>• Understand how decomposition can be used to break down problems into more manageable components</li> <li>• Be able to break down a large Computing problem into its parts and understand:               <ul style="list-style-type: none"> <li>– how images are converted to binary using pixels</li> <li>– how text is converted to binary using ASCII</li> </ul> </li> </ul> <p><b>Some students will be able to:</b></p> <ul style="list-style-type: none"> <li>• Understand how nested loops can be used to improve solutions further</li> <li>• Be able to use an algorithm to communicate data               <ul style="list-style-type: none"> <li>– Use a binary tree to further improve the algorithm</li> </ul> </li> <li>• Understand network (graph) theory terms including:               <ul style="list-style-type: none"> <li>– Nodes</li> <li>– Edges</li> </ul> </li> <li>• Be able to break down a large Computing problem into its parts and understand:               <ul style="list-style-type: none"> <li>– how data is broken up into packets</li> <li>– how data is sent through a network</li> </ul> </li> </ul>

## Curriculum Plan KS3 – 2021/2022

<b>Skills to be Gained</b>	<p>This unit introduces students to the world of computational thinking and logic. With the help of many unplugged activities, students get to understand the power of problem solving and the different methods that Computer Scientists use to tackle problems. All activities that can be carried out by computer have a paper alternative.</p> <p>This unit includes many novel activities to introduce key topics. For example, logical deductions and logical puzzles are used to show logical thinking, water pipes are used to introduce logic gates, network topology is used to show how mazes can be solved and phone messaging is used to demonstrate decomposition.</p>
<b>Assessment</b>	<p>Homework is given for each lesson. These consist of a mixture of short, factual questions and longer problems that need to be solved. A final multi-choice test of 40 questions is also given</p>
<b>Links with Prior/Subsequent Learning</b>	<p>Students should have studied algorithms and computational thinking topics either at Key Stage 2 or Key Stage 3. Some knowledge or experience with Scratch will be advantageous but is not essential. This unit has been written to be accessible for years 7-9. The unit assumes no specific prior knowledge, however some of the tasks can be challenging. It will depend on the experience and ability of students as to which year group is most appropriate for teaching this unit.</p>
<b>Numeracy/Literacy Skills</b>	<p>Vocabulary associated with this unit, such as:</p> <p>Logical thinking, logic, Boolean operators, AND, OR, NOT, logic gates, AND gate, OR gate, NOT gate, algorithm, sequence, Venn diagram, truth table, circuit, loop, nested loop, instructions, binary tree, abstraction, network, decomposition, pixels, ASCII, nodes, edges, packets, source, destination.</p>
<b>NC Links</b>	<ul style="list-style-type: none"> <li>• Design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems</li> <li>• 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</li> <li>• 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</li> <li>• 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]</li> </ul>

## Curriculum Plan KS3 – 2021/2022

Topic	YEAR 9 – CYCLE 2 INTRODUCTION TO PYTHON	
<b>Learning Objectives</b>	<p><b>At the end of this Unit all pupils should be able to:</b></p> <ul style="list-style-type: none"> <li>• Run simple Python programs in Interactive and Script mode</li> <li>• Write pseudocode to outline the steps in an algorithm prior to coding</li> <li>• Write programs using different types of data (e.g. strings and integers)</li> <li>• Correctly use different variable types (e.g. integer and floating point), assignment statements, arithmetic operators</li> <li>• Distinguish between syntax and logic errors and be able to find and correct both types of error</li> <li>• Use relational operators to control the order in which program statements are executed and in what order (if and while statements)</li> </ul>	<p><b>Most pupils will be able to:</b></p> <ul style="list-style-type: none"> <li>• Write an error-free, well-documented program involving selection and iteration</li> <li>• Describe how a binary search is carried out</li> <li>• Explain the advantages of a binary search over a linear search for an ordered list</li> </ul> <p><b>Some pupils will be able to:</b></p> <ul style="list-style-type: none"> <li>• Devise their own algorithms to solve reasonably complex problems, e.g. a binary search</li> <li>• Test and debug their programs, and correct both syntax and logic errors</li> <li>• Make allowances in their programs for user input errors, ensuring that the program still runs to a successful conclusion</li> </ul>
<b>Skills to be Gained</b>	<p>The unit is an introduction to Python, a powerful but easy-to-use high-level programming language. Although Python is an object-oriented language, at this level the object-oriented features of the language are barely in evidence and do not need to be discussed. The focus is on getting pupils to understand the process of developing programs, the importance of writing correct syntax, being able to formulate algorithms for simple programs and debugging their programs.</p>	
<b>Assessment</b>	<p>Pupils will write and run a program and submit the code and screenshots of the program running in a learning Portfolio.</p>	
<b>Links with Prior / Subsequent Learning</b>	<p>No previous learning is necessary with this unit. Pupils may have had some experience of using variables and with a variety of relational operators such as <b>If</b> and <b>Repeat</b> in graphical block-based languages such as Scratch. Applying this knowledge will help their understanding of a text-based language such as Python.</p>	
<b>Numeracy/Literacy Skills</b>	<p>Vocabulary associated with programming and particularly Python, such as: Integrated development, IDLE, interactive mode, Script mode, variable, string, syntax, assignment statement, augmented assignment operator, data type, integer, float, round, BIDMAS, selection, iteration, syntax error, logic error, debug, binary search</p>	
<b>NC Links</b>	<ul style="list-style-type: none"> <li>• Use two or more programming languages, one of which is textual, to solve a variety of computational problems; make appropriate use of data structures; design and develop modular programs that use procedures and functions</li> <li>• 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.</li> </ul>	

## Curriculum Plan KS3 – 2021/2022

Topic	YEAR 9 – CYCLE 3 – SOUND EDITING IN AUDACITY	
<b>Learning Objectives</b>	<p><b>At the end of this Unit all pupils should be able to:</b></p> <ul style="list-style-type: none"> <li>• explain how sound is digitized</li> <li>• use input and output devices to record and play sounds</li> <li>• select suitable materials for a project</li> <li>• use basic editing techniques to produce a sound file</li> <li>• work collaboratively to give and receive feedback on work done by others</li> </ul>	<p><b>Most pupils will be able to:</b></p> <ul style="list-style-type: none"> <li>• select appropriate material for a specific audience</li> <li>• combine speech, music and sound effects from different sources into one end product</li> <li>• use more sophisticated editing techniques</li> <li>• explain how their product meets the given brief</li> </ul> <p><b>Some pupils will be able to:</b></p> <ul style="list-style-type: none"> <li>• plan and create a project with the minimum of assistance</li> </ul> <p>include a range of suitable techniques and effects to produce an effective product that meets specification</p>
<b>Skills to be Gained</b>	In this unit pupils will learn how sound is digitized and stored on computers. They will learn basic sound editing techniques and how to add sound effects and mix tracks. Ways of creating different sound effects (the job of a “foley artist”) are described.	
<b>Assessment</b>	Assessment will be by means of an Assessment Portfolio, to include a description, critical review and evidence of an advertisement planned and recorded by the pupil, and a self-evaluation.	
<b>Links with Prior/Subsequent Learning</b>	Pupils should be aware that computers work digitally, and use numbers rather than analogue data. Pupils should be aware from science lessons that sound is transmitted by waves.	
<b>Numeracy/Literacy Skills</b>	<p><i>Vocabulary associated with sound editing, such as:</i> trim, effects, noise, pitch, envelope, ducking, import, export, mono, stereo, audio track, time shift</p> <p><i>Vocabulary associated with sound storage, such as:</i> sampling, frequency, amplitude, wavelength, sound wave, sampling frequency, digital, bitrate, diegetic and non-diegetic sound</p>	
<b>NC Links</b>	<ul style="list-style-type: none"> <li>• Understand how data of various types (including sounds) can be represented and manipulated digitally, in the form of binary digits</li> <li>• 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</li> </ul>	

## Curriculum Plan KS3 – 2021/2022

Topic	YEAR 9 – CYCLE 4 DATABASE DEVELOPMENT	
<b>Learning Objectives</b>	<p><b>At the end of this Unit all pupils should be able to:</b></p> <ul style="list-style-type: none"> <li>• Give examples of databases used by organisations which are accessible to the public via the Internet</li> <li>• Create a database table using several fields with different data types</li> <li>• State the purpose of a primary key in a database</li> <li>• Create a basic input form to input data</li> <li>• Query the database using more than one criterion to find answers to user queries</li> <li>• Create a basic report with suitable headings</li> <li>• Create a front-end application menu with buttons linking to a form and a report</li> </ul>	<p><b>Most pupils will be able to:</b></p> <ul style="list-style-type: none"> <li>• Add features to an input form to make it more user-friendly</li> <li>• Fully customise their input forms and reports</li> </ul> <p><b>Some pupils will be able to:</b></p> <ul style="list-style-type: none"> <li>• Create the relationship between two linked tables</li> <li>• Create a complex query which uses two tables in a relational database</li> <li>• Create a report which uses data from linked tables</li> <li>• Edit a report structure and add subtotals and/or a total to the report</li> </ul>
<b>Skills to be Gained</b>	<p>It is a practical unit covering the basic theory, creation and use of a single-table database and a simple relational database involving two tables in a one-to-many relationship. Pupils will start by looking at an existing single-table database, learning how to add records and make queries. In subsequent lessons they will create:</p> <ul style="list-style-type: none"> <li>• a flat-file or two-table relational database of their own, using suitable field types and adding in appropriate validations</li> <li>• an input form with help text, combo boxes and list boxes</li> <li>• queries and a report using data from one or both tables</li> <li>• a front end menu for their application linking to the database input form and report</li> </ul>	
<b>Assessment</b>	Pupils will create an Assessment Portfolio	
<b>Links with Prior / Subsequent Learning</b>	No previous learning is necessary with this unit. Many pupils may have a basic understanding of databases from previous years.	
<b>Numeracy/Literacy Skills</b>	Vocabulary associated with this Unit, such as: Flat-file database, relational database, table, column, record, field, query, parameter, criterion, criteria, primary key, linked tables	
<b>NC Links</b>	<ul style="list-style-type: none"> <li>• understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems</li> </ul>	

## Curriculum Plan KS3 – 2021/2022

Topic	YEAR 9 – CYCLE PRACTICAL PROGRAMMING	
<b>Learning Objectives</b>	<p><b>At the end of this Unit all students should be able to:</b></p> <ul style="list-style-type: none"> <li>• Use basic programming structures of sequence, selection and iteration</li> <li>• Use a regular expression to validate an input</li> <li>• Create, manipulate and interrogate lists</li> <li>• Read from, write to and append simple text files</li> <li>• Write, test and debug programs to solve simple problems</li> </ul>	<p><b>Most students will be able to:</b></p> <ul style="list-style-type: none"> <li>• Write error-free, well-documented programs</li> <li>• Use modular programming techniques to break down a problem into its component parts and write well-structured programs using separate functions called from a main program</li> </ul> <p><b>Some students will be able to:</b></p> <ul style="list-style-type: none"> <li>• Write error-free programs that extend advanced techniques (e.g. using 2D arrays)</li> </ul>
<b>Skills to be Gained</b>	<p>This is a theoretical unit covering the basic principles and architecture of local and wide area networks. Pupils will learn that the World Wide Web is part of the Internet, and how web addresses are constructed and stored as IP addresses. Client-server, peer-to-peer networks and the concept of cloud computing are all described. Ways of keeping data secure and simple encryption techniques are also covered</p>	
<b>Assessment</b>	<p>Assessment of this work is intended to be formative throughout, with a variety of worksheets and homework tasks to provide feedback to the teacher.</p>	
<b>Links with Prior / Subsequent Learning</b>	<p>Although this unit assumes that students have had some exposure to programming in Python, all the basics are covered and students with no experience should quickly be able to catch up.</p> <p>Students should have some previous experience of Python programming, including basic inputs and outputs, converting inputs to numbers, selection (if-elif-else statements), loops (both while and for loops) and modules (e.g. random numbers). These skills may have been acquired through the units 'Introduction to Python' and 'Python: Next Steps'</p>	
<b>Numeracy/Literacy Skills</b>	<p>Vocabulary associated with programming and particularly Python, such as:</p> <p>Integrated development, IDLE, variable, string, syntax, assignment statement, data type, integer, float, round, BIDMAS, selection, iteration, regular expression, list, two-dimensional list, text file, syntax error, logic error, debug, procedure, function, call, argument, parameter.</p>	
<b>NC Links</b>	<ul style="list-style-type: none"> <li>• Use two or more programming languages, one of which is textual, to solve a variety of computational problems; make appropriate use of data structures; design and develop modular programs that use procedures and functions</li> </ul>	