

Year 9 Cycle 1 : Computational Thinking & Logic
Reading: <https://www.bbc.co.uk/bitesize/topics/z7tp34j>

Name:		TARGET					
HWK Mark		Assessment Result:	SBE	BE	E	AE	SAE

KEYWORDS			Unit Description			
Logical thinking	Sequence	network	<p>This unit introduces students to the world of computational thinking and logic. Students get to understand the power of problem solving and the different methods that Computer Scientists use to tackle problems.</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> <p>Homework is given for each lesson. These consist of a mixture of short, factual questions and longer problems that need to be solved.</p> <p>Assessment: Takes place in the form of a final multiple-choice test of 40 questions.</p>			
Logic	Venn diagram	decomposition				
Boolean operators	truth table	pixels				
AND, OR, NOT	circuit	ASCII				
logic gates	loop	Nodes				
AND gate	nested loop	Edges				
OR gate	instructions	Packets				
NOT gate	binary tree	Source				
Algorithm	abstraction	Destination.				
NATIONAL CURRICULUM LINK(s)						
<ul style="list-style-type: none"> 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 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] 						

Please Tick off Criteria that you have met during this cycle of work		✓
7-9	Understand how networks are used to make an abstraction of a maze	
	Understand how decomposition can be used to break down problems into more manageable components	
	Be able to break down a large Computing problem into its parts and understand: how images and text are converted to binary	
6-7	Understand how abstractions are used in everyday life	
	Understand how Boolean operators can be represented in written expressions and Venn diagrams	
	Understand how logic is used in different situations	
5	Be able to complete truth tables for logic gates and circuits with up to three inputs	
	Understand how loops can be used to reduce the amount of code required for a solution	
	Be able to refine algorithms to reduce the number of instructions required	
	Understand the difference between lossy and lossless compression	
	Create a sequence of instructions to achieve a goal	
4	Know the different logic gates including: AND OR & NOT	
	Know the common Boolean operators: AND OR & NOT	
	Be able to ask logical questions to solve problems	

PREVIOUS LEARNING

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 assumes no specific prior knowledge

Year 9: Cycle 2: Introduction to Python
Reading: <https://www.bbc.co.uk/bitesize/guides/zts8d2p/revision/1>

Name:					TARGET		
HWK Mark		Assessment Result:	SBE	BE	E	AE	SAE

KEYWORDS			Unit Description				
data type	logic error	IDLE	<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. The pupils' final programs are put into a learning portfolio with evidence of correct running, for assessment purposes.</p> <p>Homework is given for each lesson. These consist of a mixture of short, factual questions assessing knowledge in isolation and longer questions in which students are asked to analyse a situation or justify their answer to questions.</p> <p>For the assessment Pupils will write and run a program and submit the code and screenshots of the program running in a learning Portfolio.</p>				
integer	debug	interactive mode					
float	binary search	Script mode					
round	Variable	String					
BIDMAS	Syntax	Sequence					
Selection	Iteration	syntax error					
NATIONAL CURRICULUM LINK							
<ul style="list-style-type: none"> 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 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. 							

Please Tick off Criteria that you have met during this cycle of work		✓
7-9	Devise their own algorithms to solve reasonably complex problems, e.g. a binary search	
	Test and debug their programs, and correct both syntax and logic errors	
	Make allowances in their programs for user input errors, ensuring that the program still runs to a successful conclusion – which may include printing an error message and stopping the run	
6 - 7	Write an error-free, well-documented program involving selection and iteration	
	Explain the advantages of a binary search over a linear search for an ordered list	
	Distinguish between syntax and logic errors and be able to find and correct both types of error	
	Use relational operators to control the order in which program statements are executed and in what order (if and while statements)	
5	Describe how a binary search is carried out	
	Correctly use different variable types (e.g. integer and floating point), assignment statements, arithmetic operators	
	Write pseudocode to outline the steps in an algorithm prior to coding	
4	Write programs using different types of data (e.g. strings and integers)	
	Use comments to document their programs and explain how they work	
	Run simple Python programs in Interactive and Script mode	

PREVIOUS LEARNING

*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 **If** and **Repeat** in graphical block-based languages such as Scratch. Applying this knowledge will help their understanding of a text-based language such as Python.*

Cycle 3 Year 9 : Sound Editing in Audacity
Reading: <https://www.bbc.co.uk/bitesize/guides/z3m39j6/revision/4>

Name:		TARGET					
HWK Mark		Assessment Result:	SBE	BE	E	AE	SAE

KEYWORDS			Unit Description			
<u>Vocabulary associated with sound editing</u>			<p>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. Pupils will undertake a creative project to analyse, plan, record and edit a short sound file. This could take the form of a radio advertisement or short podcast</p> <p>Homework is given for each lesson. These consist of a mixture of short, factual questions assessing knowledge in isolation and longer questions in which students are asked to analyse a situation or justify their answer to questions.</p> <p>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.</p>			
Trim	Envelope	Mono				
Effects	Ducking	stereo				
Noise	Import	audio track				
Pitch	Export	time shift				
<u>Vocabulary associated with sound storage:</u>						
Sampling	Frequency	digital				
Amplitude	Wavelength	bitrate				
sound wave	sampling frequency	diegetic and non-diegetic sound				
NATIONAL CURRICULUM LINK						
<ul style="list-style-type: none"> Understand how data of various types (including sounds) 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 						

Please Tick off Criteria that you have met during this cycle of work		✓
7-9	include a range of suitable techniques and effects to produce an effective product that meets specification	
	plan and create a project with the minimum of assistance	
6 - 7	use more sophisticated editing techniques	
	explain how their product meets the given brief	
	combine speech, music and sound effects from different sources into one end product	
5	select appropriate material for a specific audience	
	work collaboratively to give and receive feedback on work done by others	
	use basic editing techniques to produce a sound file	
4	select suitable materials for a project	
	explain how sound is digitized	
	use input and output devices to record and play sounds	

PREVIOUS LEARNING

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.

Cycle 4 Year 9 : Database Development

Reading: <https://www.bbc.co.uk/bitesize/guides/zswnb9q/revision/1>

Name:		TARGET					
HWK Mark		Assessment Result:	SBE	BE	E	AE	SAE

KEYWORDS			Unit Description	
Flat-file database	record	criterion	<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"> a flat-file or two-table relational database of their own, using suitable field types and adding in appropriate validations an input form with help text, combo boxes and list boxes queries and a report using data from one or both tables a front end menu for their application linking to the database input form and report <p>MS Access is used in this unit. Homework is given for each lesson. These consist of a mixture of short, factual questions assessing knowledge in isolation and longer questions in which students are asked to analyse a situation or justify their answer to questions.</p>	
relational database	field	criteria		
table	query	primary key		
column	parameter	linked tables		
NATIONAL CURRICULUM LINK				
<p>understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems</p>				

Please Tick off Criteria that you have met during this cycle of work		✓
7-9	Edit a report structure and add subtotals and/or a total to the report	
	Create a report which uses data from linked tables	
	Create a complex query which uses two tables in a relational database	
6 - 7	Create the relationship between two linked tables	
	Add features to an input form to make it more user-friendly	
	Fully customise their input forms and reports	
5	Create a front-end application menu with buttons linking to a form and a report	
	Create a basic report with suitable headings	
	Query the database using more than one criterion to find answers to user queries	
	Create a basic input form to input data	
4	State the purpose of a primary key in a database	
	Create a database table using several fields with different data types	
	Give examples of databases used by organisations which are accessible to the public via the Internet	

Prior Learning

No previous learning is necessary with this unit.

KS3 : Practical Programming in Python
Reading: <https://www.bbc.co.uk/bitesize/topics/zhv39j6>

Name:		TARGET					
HWK Mark		Assessment Result:	SBE	BE	E	AE	SAE

KEYWORDS			Unit Description			
IDLE	Round	Syntax error	<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 are given numerous examples of how to write Python programs to solve different types of problem along with some of the most common programming techniques used, such as validating data entry and creating a menu system with separate functions for each of the menu options.</p> <p>Homework is given for each lesson. These consist of a mixture of short, factual questions assessing knowledge in isolation and longer questions in which students are asked to analyse a situation or justify their answer to questions.</p> <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>			
Variable	BIDMAS,	Logic error				
String	Selection	Debug				
Syntax	Iteration	Procedure				
Assignment statement	Two-dimensional list	Regular expression				
Data type	List	Call				
Integer	Function	Argument				
Float	Text file	Parameter				
NATIONAL CURRICULUM LINK						
<ul style="list-style-type: none"> 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 						

Please Tick off Criteria that you have met during this cycle of work		✓
7-9	Write error-free programs that extend advanced techniques (e.g. using 2D arrays)	
	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	
6 - 7	Write error-free, well-documented programs	
	Write, test and debug programs to solve simple problems	
5	Read from, write to and append simple text files	
	Create, manipulate and interrogate lists	
4	Use a regular expression to validate an input	
	Use basic programming structures of sequence, selection and iteration	
	Create simple Input / Output messages	

PREVIOUS LEARNING

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'