Computer Science Unit Overview Year 9

Subject not previously studied. Students will follow planned scheme of work.

Computer Science - Year 9 Half Term 1 0.1 Using OneNote and 7.1 Visual Scratch Programming				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**What does mastery look like?**	**How does this build on prior learning?**	**What additional resources are available?**
0.1Using OneNote 7.1 Visual Scratch Programming	**Knowledge**: This unit of work teaches an introduction to programming using the Scratch programming language. Students will be introduced to programming inputs, variable storage, outputs, sequencing and selection. **Understanding:** How to input values into scratch Where inputs are stored in scratch How to output values onto the screen in scratch How programs make decisions How computers may use more than 1 variable How operators work in scratch How Flowcharts work **Skills:** Constructing programs from a given brief Using Modern Technology for Learning Using Graphic Programming Tools	Identify code that will demand the user to enter an input Identify code that will output data. Identify code that will store data Construct a program from a flow chart that will ask the user for information and respond using the entered information Construct a program from a flow chart that will ask the user for two numbers, add them together and display the result. Construct a program from a flow chart that will ask the user for two numbers, decide which is the biggest number and display it. Construct a program which will ask the user for a score out of 100. If the user enters a score of 50 or higher the message 'You passed' is displayed. If a score of less than 50 is enterded the message 'You failed' is displayed.	Introduces students to OneNote/ClassNotebook. Develops problem solving skills through a graphic programming language.	Scratch support https://uploads.scratch.mit.edu/about/

Computer Science - Year 9 Half Term 2 7.6 Micro:Bit Madness				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**What does mastery look like?**	**How does this build on prior learning?**	**What additional resources are available?**
7.6 Micro:Bit Madness	**Knowledge**: This unit introduces students to the Micro:Bit device and teaches them how to program a variety of applications including a digital dice, digital compass and games console (pong). The unit uses both the 'Blocks' and 'Python' programming language. **Understanding:** Identify parts of a Micro:Bit Steps needed to create a programmed solution to a problem **Skills:** Code programs that make use of the sensors Test Programs on the virtual Micro:Bot Able to transfer a program to the Micro:Bit	Write a Micro:Bit program which counts Write a Micro:Bit program which uses random selection Write a Micro:Bit program which uses a forever loop and motion sensor	KS2 NC design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts KS4 NC develop their capability, creativity and knowledge in computer science, digital media and information technology Practical application of graphic programming skills from 7.1 Visual Scratch Programming	https://www.microbit.co.uk/blocks/lessons/

Computer Science - Year 9 Half Term 3 8.3 Introduction to Python				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**What does mastery look like?**	**How does this build on prior learning?**	**What additional resources are available?**
8.3 Introduction to Python	**Knowledge**: Data types in python, integer, string and real **Understanding:** Where inputs are stored in python How python can do calculations How computers can make a decision, if statements in python **Skills:** Outputting text to the screen How to output values onto the screen in python, variables	Write a python program which asks for an input and displays it back on the screen Write a python program which performs a mathematical calculation Write a python program which compares two values	KS2 NC use sequence, selection, and repetition in programs; work with variables and various forms of input and output KS4 NC develop and apply their analytic, problem-solving, design, and computational thinking skills Develops problem solving skills from 7.1 and 7.6 in a text based programming language.	https://www.python.org/about/help/

Computer Science - Year 9 Half term 4 8.2 Binary Bits and Bobs				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**What does mastery look like?**	**How does this build on prior learning?**	**What additional resources are available?**
8.2 Binary Bits and Bobs	**Knowledge**: Know why computers use the binary number system ASCII code **Understanding:** Understand the binary number system and why it is important in computing **Skills:** Be able to read a binary number and work out its value Be able to write a denary number in binary Be able to add binary numbers together Be able to add two binary numbers together by converting to denary and then adding Be able to add two binary numbers without converting first Be able to convert denary to binary and then add them together	Be able to convert binary to denary Be able to convert denary to binary Be able to convert from binary to denary and work out corresponding letter from ASCII table	**KS3 NC** 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	

Computer Science - Year 9 Half Term 5 9.6 Computational Thinking				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**What does mastery look like?**	**How does this build on prior learning?**	**What additional resources are available?**
9.6 Computational Thinking	**Knowledge**: Decomposition Pattern recognition Writing instructions (algorithms) Flow charts Pseudocode Evaluating algorithms **Understanding:** Understand that by breaking down a large problem it can become simpler to solve **Skills:** Be able to break down a problem into manageable sections Be able to break down a problem and find alternative solutions for small sections of the overall problem Be able to spot where sections of a problem are repeated (pattern recognition) Be able to write instructions which reduce repetition unnecessarily Be able to transfer ideas and solutions from one problem to another Abstract data and understand why it is important to ignore irrelevant information	Describe the key words decomposition, pattern recognition, abstraction, algorithm and iteration Identify iteration type in flowcharts Turn PseudoCode to Flow Create Pseudocode for a problem	Develops problem solving skills from 7.1 Visual Scratch, 7.6 Micro:Bit Madness and 8.3 Introduction to Python.	cs4fn, Queen Mary, University of London (www.cs4fn.org)

Computer Science - Year 9 Half Term 6 9.1 Python programming				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**What does mastery look like?**	**How does this build on prior learning?**	**What additional resources are available?**
9.1 Python programming	**Knowledge**: Data types (integer, string, real), and working with numbers **Understanding:** Iteration in python, for and while loops Starting number, upper limit and steps in for loops **Skills:** Outputting text to the screen Storing inputted data into a variable # Commenting on code	Write programs that: Display a message to the screen Asks the user for their favourite colour and then comments on the colour entered (for example, *"**colour entered***, is a nice colour!" Asks the user for their age and then displays what their age will be in 50 years. Asks the user for the dimensions for a box and then works out its volume Asks the user for their weekly pocket money, weekly phone bill, money spent on food each week and money spent on seeing friends each week. The program is to then display how much pocket money will be left when the week is over.	8.3 Introduction to python	Python code for kids is a clearly written summary of the Python language written in accessible language: www.pythondictionary.code-it.co.uk/ 'Python in 10 minutes' is a quick run through of the basic concepts: www.korokithakis.net/tutorials/python/