Year 8			***Curriculum Checkpoints: What do students know and what can they do?***			
Computing			Developing	Securing	Mastering	Excelling
AF1	Digital Graphics - Photoshop	Knowledge	Can understand the terms target audience and visualisation diagram. Can draw a simple visualisation diagram.	Can create a detailed visualisation diagram for a target audience.	Can create a detailed visualisation diagram showing target audience characteristics.	Can create a detailed visualisation based on a given target audience and can fully justify elements based on target audience characteristics.
		Practical Skills	Will create a film poster in Photoshop by adding text and images.	Will create a film poster in Photoshop using basic editing tools.	Will create a film poster in Photoshop using both basic and advanced editing tools.	Will create a near professional film poster in Photoshop using advanced editing tools which fully meets the requirements of the brief and target audience.
AF2	Computer Systems & Data Representation	Knowledge	Is able to identify input and output devices and some parts of a computer. Can understand the term fetch-decode-execute cycle. Can define an embedded system. Can define the terms, computer network, local area network and wide area network.	Is able to explain how input and output devices work together. Can explain three or more parts of a computer. Can explain the term fetch-decode-execute cycle. Can give examples of an embedded system. Identifies and explains examples of local area networks and wide area networks.	Is able to identify which peripherals are input and output devices. Can explain five or more parts of a computer and how they work together. Can explain the term fetch-decode-execute cycle and how it links with the CPU and RAM. Can differentiate between an embedded system and a microprocessor. List pros and cons of using a network with suitable examples.	Is able to explain the functionalities of a range of input and output devices. Can physically identify and explain all parts of an internal computer and how they work together. Can explain the term fetch-decode-execute cycle and how it links with the CPU and RAM and give a suitable example of the process. Can explain the benefits and drawbacks of using embedded systems. Know the features of bus, ring and star network topologies.
		Practical Skills	Identifies the pattern in the binary number sequence which is needed to convert 4 bit and 8 bit binary. Can identify numbers used in base 2 and 10 and know the difference between them. Can draw the and, or and not gates. Can draw a visual representation of a LAN and WAN.	Converts 4 bit binary into denary and convert a denary number into 4 bit binary. Can show working out for the conversions. Can explain the different rules for each logic gate. Can identify which network has been used in a given scenario.	Identifies a bit, nibble and byte in binary and the difference between them. Can start to convert 8 bit binary into denary. Can work out the largest denary number made with 4 bit and 8 bit binary. Can complete a truth tables for each logic gate and work out outputs. Give examples of when they have used a network and assess how useful it was.	Describes a bit, nibble and byte in binary. Can convert 8 bit binary into denary and convert a denary number into 8 bit binary. Can convert denary and binary numbers into hexadecimal. Can work out outputs of a combination of two or more logic gates. Draw bus, ring and star topologies.
AF3	Digital Artefacts - Spreadsheets	Knowledge	Is able to identify some key parts of a spreadsheet. Can understand that a formulae begins with an = symbol.	Explains some key parts of a spreadsheet. Can understand the difference between profit and loss. Can understand and write the correct structure for a formulae.	Explains most key parts of a spreadsheet. Can explain the difference between the terms profit and loss. Can correctly write out formulae and functions for different spreadsheet scenarios.	Explains all parts of a spreadsheet and use them appropriately. Can explain how a spreadsheet is used in business to calculate profit or loss. Can write and explain formulae and functions.
		Practical Skills	Uses some basic formatting tools on a spreadsheet. Can use simple formulae for calculations. Can use the SUM function on a spreadsheet. Can create a graph/chart on a spreadsheet. Can create a basic spreadsheet using formulae.	Uses a wide range of formatting tools on a spreadsheet. Can use formulae to work out profit and loss. Can use some advanced functions on a spreadsheet. Can create and correctly label different types of graphs and charts.	Creates a spreadsheet which is correctly formatted and uses a range of formulae and advanced functions to perform calculations. Can present information from a spreadsheet in a range of different graphs. Can use conditional formatting on a spreadsheet.	Creates a spreadsheet which uses a wide range of formulae and advanced functions to perform calculations. Can justify the choices of graphs and charts created in a spreadsheet. Can explain the purpose and of use conditional formatting in a spreadsheet.

AF4	**Text based Programming Languages - Python**	**Knowledge**	Can define the terms input, output, variable and conditional statement and comparison operators.	Can identify Python code which uses input, output, variable, loops, conditional statement and comparison operators. Can spot syntax errors in Python code.	Can explain lines of Python code which uses input, output, variable, loops, conditional statement and comparison operators. Can spot syntax and logic errors in Python code.	Can justify why conditional statements and loops are used. Can spot and correct syntax and logic errors in Python code.
		Practical Skills	Writes lines of code in Python which include; inputs, outputs and variables	Implements calculator program which includes, inputs, outputs and variables. Can explain why an integer data type is important. Can define what a conditional statement is.	Implements a calculator program which includes, inputs, outputs and variables. Can explain why an integer data type is important and implement it in the calculator program. Can define and implement a conditional statement.	Creates Python programs independently using inputs, outputs, variables, conditional statements, arithmetic and comparison operators. Can implement loops into their programs.