IT/Computing

ALGORITHMS	Foundation (1-3)	Developing (4-5)	Secure (6-7)	Excellence (8-9)
Year 7	Students have some idea what an algorithm is • Can recognise common sequences of instructions as examples of algorithms. • Can identify that goals can be achieved following a sequence of steps. • Understand that a computer programme is a sequence of instructions written to perform a specified task with a computer.	Students have a good idea what an algorithm is • Understand that Algorithms are sets of instructions for achieving goals, made up of pre-defined steps. • Can create a simple algorithm to solve a problem.	Students have a very good idea what an algorithm is • Can use the correct shapes for processes and input/outputs commands. I can appreciate that some algorithms are more efficient than others. • Can think about everyday algorithms, such as classroom rules or procedures, and look for easier or faster ways to get things done. • Can use IF and Else/ELIF statements within my algorithms.	Students are clear of what an algorithm is • Can create an algorithm using complex selection and iteration commands using the correct shapes. • Can explain an algorithm in detail and understand the problem it is trying to solve.
Year 8	• Can take a problem and divide it into all its sub-problems and show this as a diagram. • Can recognise common sequences of instructions as examples of algorithms. • Can recognise some programming techniques used in algorithms.	• Can create a simple model for a complex problem. • Can use the correct shapes for processes and input/outputs commands. • Can use IF and Else/ELIF statements within my algorithms.	• Can take a problem and divide it into all its sub-problems and show this as a diagram. • Can recognise key programming techniques used within Algorithms, including selection and iteration. • Understand that programmes can use math operators and mainly correctly utilise them in algorithms.	• Can create an accurate, detailed model for a complex problem. • Can give carefully reasoned explanations of what a programme will do under given circumstances, including some attempt at explaining why it does what it does. • Understand that programmes can use math operators and correctly utilise them in algorithms.

IT/Computing

GRAPHICS	Foundation (1-3)	Developing (4-5)	Secure (6-7)	Excellence (8-9)
Year 7	• Can research, select, edit and use information from given digital sources. • Can create and edit text onscreen, combining images and/or sound. • Can talk or use a writing frame of keywords to state how to improve my work.	• Can research and plan for a new digital media project in response to a brief, using a storyboard. • Can make modifications to improve my work based on feedback. • Can explain the tools I have used to create my project.	• Can process found or self-produced assets, including text, data, sound, still or moving images, and combine these to create, present and communicate their work, showing an awareness of audience and purpose. • Can explain the difference between a vector and bitmap image.	• Can manipulate and integrate a combination of text, data, sound, still and moving images, to create a multimedia product, for specific audience. • Can increase the smoothness of animations by using a higher frame rate.
Year 8	• Can research and plan for a new digital media project in response to a brief, using a storyboard. • Can change the speed of animations to suit the given project. • Can make modifications to improve work based on feedback.	• Can create new images with inbuilt tools or import an existing image. • Can add existing sounds or choose from the programme's library. • Can evaluate work and make judgements on how it fits the target audience.	• Can manipulate and integrate a combination of text, data, sound, still and moving images, to create a multimedia product, for specific audience. • Can increase the smoothness of animations by using a higher frame rate. • Can create layers to separate the different bits of an animation. • Can evaluate work and make judgements on how it fits the target audience, the brief and describe changes could make to improve.	• Can effectively use a range of appropriate software facilities, which includes digital video, web and multimedia authoring software, to produce a solution which meets user needs. • Can use a timeline to keep track of my animation. • Can evaluate work and describe how it fits with the target audience and why. • Can explain and demonstrate changes to improve my work.

IT/Computing

PROGRAMMING	Foundation (1-3)	Developing (4-5)	Secure (6-7)	Excellence (8-9)
Year 7	• Can write programmes that accomplish specific goals. • Understand that a computer programme is a sequence of instructions written to perform a specified task with a computer. • Can create a programme by entering instructions that are executed in a sequence.	• Can create a simple programme on screen, correcting any errors. • Can write programmes that use sequence, selection, repetition and variables in programmes. • Understand and use the IF selection block in coding challenges.	• Understand and can use IF and Else/ELIF statements. • Can demonstrate using relational operators within my code. • Understand and use While loops effectively in coding challenges. • Can read snippets of code and understand what the output would be.	• Understand and use selection (both IF... and IF...ELSE…) effectively in coding challenges. • Understand and use a range of comparison operators in both selection and iteration effectively. • Understand and use a range of Boolean operators in both selection and iteration effectively. • Can apply understanding of mathematical concepts effectively to programmatically.
Year 8	• Can make predictions over what a programme will do. • Can identify some programming techniques within a block of code. • Write code with sequence, input, variables output commands and selection statements using text based programming.	• Can plan, create, test and reflect on a solution to a problem that a computer could solve. • Understand and can use IF and Else/ELIF statements. • Can demonstrate using variables and relational operators within a while loop. • Can identify errors in code and fix some syntax errors.	• Can use comparison and logical operators within my code. • Can use iteration (While loops) in programmes and identify reasons for using loops. • Can debug programmes correcting syntax and some logical errors. • Can create programmes that use different types of data [integers, characters, strings] for text based programming.	• Can debug programmes correcting syntax and some logical errors. • Understand how instructions can be written efficiently and am able to describe the efficiency of my programmes. • Can create a programme using iteration (For and While Loops) • Can create a programme that uses string manipulation to make the programme robust.