

Computer Science Progression of Skills

| Year 1 | Year 2 | Year 3 | Year 4 | Year 5 | Year 6 |
|---|---|---|---|---|---|
| <p>I can explain that an algorithm is a set of instructions.</p> <p>I can understand that a computer program turns an algorithm into code that the computer can understand.</p> <p>I can work out what is wrong when the steps are out of order in instructions.</p> <p>I can try and fix my code if it isn't working properly.</p> <p>I can make good guesses of what is going to happen in a program.</p> | <p>I can explain an algorithm is a set of instructions to complete a task.</p> <p>I can understand I need to carefully plan my algorithm so it will work when I make it into code.</p> <p>I can design a simple program using 2Code that achieves a purpose.</p> <p>I can find and correct some errors in my program.</p> <p>I can say what will happen in a Program.</p> <p>I can spot something in a program that has an action or effect (does something).</p> | <p>I can base a written algorithm for a program upon a real-life situation.</p> <p>I can design an algorithm carefully, thinking about what I want the program to do and how I could turn my algorithm into code.</p> <p>I can design a program thinking logically about the sequence of steps required.</p> <p>I can experiment with timers in my programs.</p> <p>I can experiment with the effect of using repeat commands.</p> <p>I can identify the difference in using the effect of a timer or repeat command in my code.</p> <p>I can identify an error in my program and fix it.</p> <p>I can read programs with several steps and predict what it will do.</p> <p>I can identify different ways that the Internet can be used for communication.</p> <p>I can use email such as 2Email to respond to others appropriately and attach files.</p> | <p>I can turn a real-life situation to solve into an algorithm, using a design that shows how I can accomplish this in code.</p> <p>I can use repetition in my code. For example, using a loop that continues until a condition is met such as the correct answer being entered.</p> <p>I can use timers within my program designs more accurately to create repetition effects.</p> <p>I can use selection (decision) in my programming. For example, using an 'if statement' for a question being asked and the program takes one of two paths.</p> <p>I can use variables within my program and know how to change the value of variables.</p> <p>I can use the user inputs and output features within my program, such as 'Print to screen'.</p> <p>I can identify errors in my code by using different methods, such as stepping through lines of code and fixing them.</p> <p>I can read programs that contain several steps and predict the outcomes with increasing accuracy.</p> <p>I can recognise the main component parts of hardware which allow computers to join and form a network.</p> <p>I can understand that network and communication components can be found in many different devices which allow them to join the internet.</p> | <p>I can make more complex real-life problems into algorithms for a program.</p> <p>I can test and debug my programs as I work.</p> <p>I can convert (translate) algorithms that contain sequence, selection and repetition into code that works.</p> <p>I can use sequence, selection, repetition, and some other coding structures in my code.</p> <p>I can organise my code carefully for example, naming variables and using tabs. I know this will help me debug more efficiently.</p> <p>I can use logical methods to identify the cause of any bug with support to identify the specific line of code.</p> <p>I can understand the importance of computer networks and how they help solve problems and enhance communication.</p> <p>I can recognise the main dangers that can be perpetuated via computer networks.</p> <p>I can explain what personal information is and know strategies for keeping this safe.</p> <p>I can use the most appropriate form of online communication according to the digital content. For example, use 2Email, 2Blog and Display Boards.</p> | <p>I can turn a complex programming task into an algorithm.</p> <p>I can identify the important aspects of a programming task (abstraction).</p> <p>I can decompose important aspects of a programming task in a logical way, identifying appropriate coding structures that would work.</p> <p>I can test and debug my program as I work on it and use logical methods to identify a cause of a bug.</p> <p>I can identify a specific line of code that is causing a problem in my program and attempt a fix.</p> <p>I can translate algorithms that include sequence, selection and repetition into code and nest these structures within each other.</p> <p>I can use inputs and outputs within my coded programs such as sound, movement and buttons and represent the state of an object.</p> <p>I can interpret (understand) a program in parts and can make logical attempts to put the separate parts together in an algorithm to explain the program as a whole.</p> <p>I can explain the difference between the Internet and the World Wide Web.</p> <p>I can explain what a WAN and LAN is and describe the process of how access to the internet in school is possible.</p> |