

Progression in Programming

Upper Key Stage 2



	Year 5	Year 6
POS statements Sept 2014	<ul style="list-style-type: none"> Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts Use sequence, selection, and repetition in programs; work with variables and various forms of input and output Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs 	
Children understand concepts 	<ul style="list-style-type: none"> I can decompose a problem into smaller parts to design an algorithm for a specific outcome and use this to write a program for a device or onscreen activity. I can use a variable to increase programming possibilities. I can change an input to a program to achieve a different output. I can use 'if' and 'then' commands to select an action. I can refine a procedure using repeat commands to improve a program. I can talk about how a computer model can provide information about a physical system. I can use logical reasoning to detect and debug mistakes in a program. I use logical thinking, imagination and creativity to extend a program. 	<ul style="list-style-type: none"> I can deconstruct a problem into smaller steps, recognising similarities to solutions used before. I can explain and program each of the steps in my algorithm (for a device or onscreen activity). I can evaluate the effectiveness and efficiency of my algorithm while I continually test the programming of that algorithm. I can recognise when I need to use a variable to achieve a required output. I can use a variable and operators to stop a program. I can use different inputs (including sensors) to control a device or onscreen action and predict what will happen. I can link errors in a program to a problem in the algorithm on which it is based.
Teacher enables progress 	<ul style="list-style-type: none"> Set problem solving tasks for children where they are expected to write procedures to achieve outcomes Model the writing of an algorithm to achieve a specific outcome Model the detecting and correcting errors in a program and link this to the original algorithm Set tasks for children to use sensing equipment to control actions of a device Provide opportunities for children to simulate controlling effects of a physical system Talk about a variable as a container or box to store a number that needs to be used in a program Talk about the uses of programming in the world around and its impact on society, including that of gaming 	
Children build skills 	<ul style="list-style-type: none"> Explore procedures using repeat to achieve solutions to problems with Logo and a floor robot Talk about procedures as parts of a program Refine procedures to improve efficiency Use a variable to replace number of sides in a regular shape Explore instructions to control software or hardware with an input and using if... then... commands Explore a computer model to control a physical system Change inputs on a model to achieve different outputs Refine and extend a program Identify difficulties and articulate a solution for errors in a 	<ul style="list-style-type: none"> Record in some detail the steps (the algorithm) that are required to achieve an outcome and refer to this when programming Predict the outputs for the steps in an algorithm Increase confidence in the process to plan, program, test and review a program Write a program which follows an algorithm to solve a problem for a floor robot or other model Write a program which follows an algorithm to achieve a planned outcome for appropriate programming software Control on screen mimics and physical devices using one or more input and predict the outputs

Progression in Programming

Upper Key Stage 2



	<p>program</p> <ul style="list-style-type: none"> Group commands as a procedure to achieve a specific outcome within a program Write down the steps required (an algorithm) to achieve the outcome that is wanted and refer to this when programming 	<ul style="list-style-type: none"> Understand how sensors can be used to measure input in order to activate a procedure or sequence and talk about applications in society Create variables to provide a score or trigger an action in a game Link errors in a program to problems in the original algorithm
<p>Suggested activities for children to develop process</p> 	<p>Procedures, including those using an if command, are used with a floor robot such as Probot to plan an efficient route around a 'road system'. Lights must be turned on when the robot goes through a tunnel.</p>	<p>Floor robots such as Probots or Lego Mindstorms are used for children to program models to successfully navigate courses including the use of input devices to avoid obstacles.</p>
	<p>Textease Turtle or other logo program such as Logo in www.j2e.com/j2code, FSW logo or i-LOGO app is used to develop procedures to create specific affects on screen extending their repertoire of commands. They plan algorithms that require inputs and make use of variables to create a program. They create a procedure to draw any regular polygon using number_of_sides and length. They experiment with mathematical sequences of addition, subtraction, multiplication and division.</p>	<p>Use logo to create a square shaped spiral, a star. Create a procedure to calculate the area of a rectangle, the area of a circle. Use print and readword to create function machines for mathematical calculations such as the area of different quadrilaterals.</p> <p>The Blockly maze provides a useful assessment of children's understanding of both logo and planning and using an algorithm. Other Blockly resources can be used to develop logical thinking.</p>
	<p>Scratch, Visual programming in www.j2e.com/j2code or Tynker is used to develop games to meet specific challenges such as a tennis game (a version is available without the variables to support learners who need an additional step before adding variables). A score is added to the game. A picoboard is used with Scratch (download) to trigger actions based on the sensing of different variables such as sound. Different planning sheets are available to support the creation of new games.</p>	<p>Children create a Times Tables game making use of a variable. They adapt it to create a quiz where the correct answer must be linked to a question. The use of a picoboard is developed for children to use different controls for their games. Children use what they learn from examples of games to create their own. Children create their own blocks which are procedures to achieve specific outcomes.</p>
	<p>Flowol 4 or Go Control is used to program a sequence of actions for a Zebra Crossing, a light house, lights and a foundation for the Town Hall during the day and the night, and bridge lights. These can be linked to models where these are available.</p>	<p>Flowol or Go Control is used to program a sequence of actions for crossroads, pelican crossing, and the train level crossing. These can be linked to models where available.</p>
	<p>Use the Espresso Computer Modelling module to manage water at home and then the supply from a reservoir.</p>	
	<p>Kodu is introduced to plan and create games for their peers. (This is a free download but may not work on laptops unless they have a good graphics card. It seems to function without a problem on most PCs.) Increasingly complex games should be created Hopscotch app is used to develop skills, knowledge and understanding of programming as described for Scratch or Tynker above. A.L.E.X app and Cargo-Bot are used to develop problem solving skills.</p>	
<p>Sign up to Code Club to increase the repertoire of games children are creating in Scratch and Tynker. Both Scratch and Tynker provide their own tutorials and help videos. In Scratch go to help and scroll down to the bottom of the page to find a set of really useful short videos. 2DIY could be purchased and its use extended by introducing children to the program language behind the visual interface.</p>		