



Computing Progression - Programming: Create software to allow computers to solve problems

Key Stage 1	Lower Key Stage 2	Upper Key Stage 2
Across KS1 pupils will:	In lower KS2 pupils will:	In upper KS2 pupils will:
Compare forwards and backwards movements. Predict the outcome of a sequence involving forwards and backwards commands. Start a sequence from the same place. Compare left and right turns. Experiment with turn and move commands to move a robot. Predict the outcome of a sequence involving up to four commands. Follow a sequence. Plan algorithms for different parts of a task. Put together the different parts of their program test and debug each part. Compare different programming tools. Find and use commands to move a sprite. Run a program	Recognise that commands in Scratch are represented as blocks. Create a program following a design. Identify that each sprite is controlled by the commands they choose. Create a sequence of connected commands. Explain that the objects in their project will respond exactly to the code. Start a program in different ways. Combine sound commands. Build a sequence of commands. Implement their algorithm as code. Choose which keys to use for actions and explain their choices. Explain the relationship between an event and an action.	Program a microcontroller to make an LED switch on. Connect more than one output component to a microcontroller. Design sequences that use count-controlled loops. Use a count-controlled loop to control outputs. Design a conditional loop. Explain that a condition is either true or false. Program a microcontroller to respond to an input. Test and debug their project. Use selection to produce an intended outcome. Modify a condition in a program. Create a program with different outcomes using selection. Share their program with others.

Use a start block in a program	Identify a way to improve a program.	Test their program.
Use more than one block by joining them together.	Program movement.	Extend their program further.
Find blocks that have numbers.	Use a programming extension.	Identify the setup code they need in their program.
Change a value and say what happens when changed.	Choose suitable keys to turn on additional features match a piece of code to an outcome.	Identify ways the program could be improved.
Add blocks to each of their sprites.	Modify a program using a design.	Explain that the way a variable changes can be defined.
Delete a sprite	Test a program against a given design.	Identify examples of information that is variable.
Show that a project can include more than one sprite.	Create a code snippet for a given purpose.	Identify that variables can hold numbers or letters.
Add programming blocks based on their algorithm.	Explain the effect of changing a value of a command.	Identify a program variable as a placeholder in memory for a single value.
Test the programs they have created.	Program a computer by typing commands.	Make use of an event in a program to set a variable.
Identify that a program needs to be started.	Identify patterns in a sequence.	Recognise that the value of a variable can be used by a program.
Identify the start of a sequence.	Use a count-controlled loop to produce a given outcome.	Test the code that they have written.
Show how to run a program.	Use a procedure in a program.	Test their program on an emulator.
Change the outcome of a sequence of commands.	Develop my program by debugging it.	Transfer their program to a controllable device.
Match two sequences with the same outcome.	Modify a snippet of code to create a given outcome.	
Predict the outcome of a sequence of commands.		

Build the sequences of blocks they need. Create a program based on the new design. Create an algorithm. Debug their own program.	Choose when to use a count-controlled and an infinite loop. Modify loops to produce a given outcome. Recognise that some programming languages enable more than one process to be run at once. Identify which parts of a loop can be changed. Re-use existing code snippets on new sprites.	Use a variable in an if, then, else statement to select the flow of a program. Use a condition to change a variable. Modify a program to achieve a different outcome. Use an operand (e.g. <=>) in an if, then statement.
--	--	---

Computing Progression

Programming - Create software to allow computers to solve problems

Statements in red are linked to other computing strands

(NW - Networks, CM - Creating Media, DI - Data and Information, DD - Design and Development, CS - Computing Systems, IT - Impact of technology, AL - Algorithms, PG - Programming, ET - Effective use of tools, SS - Safety and security)

Key Stage 1	Years 3 and 4	Years 5 and 6
Across KS1 children will be able to: Moving a robot - compare forwards and backwards movements - predict the outcome of a sequence involving forwards and backwards commands - start a sequence from the same place - compare left and right turns - experiment with turn and move commands to move a robot - predict the outcome of a sequence involving up to four commands Robot algorithms - follow a sequence - plan algorithms for different parts of a task (AL + DD) - put together the different parts of their program - test and debug each part of the program (AL) Programming Animations	In Year 3 children will be able to: Sequencing Sounds - recognise that commands in Scratch are represented as blocks - create a program following a design - identify that each sprite is controlled by the commands they choose - create a sequence of connected commands - explain that the objects in their project will respond exactly to the code - start a program in different ways - combine sound commands - build a sequence of commands - implement their algorithm as code (AL+ET) Events and Actions in Programs - choose which keys to use for actions and explain their	In Year 5 children will be able to: Selection in physical computing - program a microcontroller to make an LED switch on - connect more than one output component to a microcontroller - design sequences that use count-controlled loops - use a count-controlled loop to control outputs - design a conditional loop - explain that a condition is either true or false - program a microcontroller to respond to an input - test and debug their project - use selection to produce an intended outcome Selection in Quizzes - modify a condition in a program - create a program with different outcomes using selection - share their program with others

• compare different programming tools • find which commands to move a sprite • use commands to move a sprite • run a program • use a Start block in a program • use more than one block by joining them together • change the value • find blocks that have numbers • say what happens when they change a value • add blocks to each of their sprites • delete a sprite • show that a project can include more than one sprite (DD) • add programming blocks based on their algorithm (AL) • test the programs I have created Programming Quizzes • identify that a program needs to be started • identify the start of a sequence • show how to run a program • change the outcome of a sequence of commands • match two sequences with the same outcome • predict the outcome of a sequence of commands • build the sequences of blocks they need • create a program based on the new design (DD) • create an algorithm (AL) • debug their own program	choices • explain the relationship between an event and an action • identify a way to improve a program • program movement (ET) • use a programming extension • choose suitable keys to turn on additional features • match a piece of code to an outcome • modify a program using a design (DD) • test a program against a given design (DD) In Year 4 children will be able to: Repetition in shapes • create a code snippet for a given purpose (AL) • explain the effect of changing a value of a command • program a computer by typing commands • identify patterns in a sequence • use a count-controlled loop to produce a given outcome • use a procedure in a program • develop my program by debugging it Repetition in games • modify a snippet of code to create a given outcome • choose when to use a count-controlled and an infinite loop • modify loops to produce a given outcome • recognise that some programming languages enable more than one process to be run at once • identify which parts of a loop can be changed • re-use existing code snippets on new sprites	• test their program • extend their program further (DD) • identify the setup code they need in their program (DD) • identify ways the program could be improved (DD) In Year 6 children will be able to: Variable in games • explain that the way a variable changes can be defined • identify examples of information that is variable • identify that variables can hold numbers or letters • identify a program variable as a placeholder in memory for a single value • make use of an event in a program to set a variable • recognise that the value of a variable can be used by a program • test the code that they have written Sensing Movement • test their program on an emulator (CS) • transfer their program to a controllable device (CS) • use a variable in an if, then, else statement to select the flow of a program • use a condition to change a variable • modify a program to achieve a different outcome • use an operand (e.g. <=>) in an if, then statement
--	---	--