

Progression Map: **Coding &** **Programming**

KS1: National Curriculum Expectations

Pupils should be taught to:

- understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions;
- create and debug simple programs;
- use logical reasoning to predict the behaviour of simple programs;
- use technology purposefully to create, organise, store, manipulate and retrieve digital content;
- recognise common uses of information technology beyond school;
- use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies.

KS2: National Curriculum Expectations

Pupils should be taught to:

- 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;
- understand computer networks including the internet; how they can provide multiple services, such as the world wide web, and the opportunities they offer for communication and collaboration;
- use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content;
- select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information;
- use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.

KS1: Coding & Programming

Children begin to understand their influence on technology by developing their programming skills to determine output. They begin to understand that an algorithm is a series of steps for solving problems and a code is a series of steps that machines can execute. They begin to explore debugging, predicting when codes may not work and changing them.

KS1 Computing National Curriculum

Children understand what algorithms are, how they are implemented as programs on digital devices, and that programs execute by following precise and unambiguous instructions. They create, debug and use logical reasoning to predict the behaviour of simple programs.

Children can:

- a give commands one at a time to control direction and movement, including straight, forwards, backwards, turn;
- b control the nature of events: repeat, loops, single events and add and delete features;
- c give a set of instructions to follow and predict what will happen;
- d improve/change their sequence of commands by debugging;
- e use key vocabulary to demonstrate knowledge and understanding in this strand: algorithm, instruction, order, debug, program, turn, left, right, clockwise, anticlockwise, blocks, sequence, project, repeat, repeat forever, invisible, grow, shrink.

LKS2: Coding & Programming

Children build on their programming skills by solving problems and programming commands to achieve a specific outcome. They begin to write programs, explain algorithms and identify errors in their work.

KS2 Computing National Curriculum

Children design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; they solve problems by decomposing them into smaller parts. They use sequence, selection, and repetition in programs and work with variables and various forms of input and output. They use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.

Children can:

- a use logical thinking to solve an open-ended problem by breaking it up into smaller parts;
- b write a program, putting commands into a sequence to achieve a specific outcome;
- c give a set of instructions to follow and predict what will happen;
- d keep testing a program and recognise when it needs to be debugged;
- e use variables to create an effect, e.g. repetition, if, when, loop;
- f use key vocabulary to demonstrate knowledge and understanding in this strand: decompose, decomposing, logical sequence, flowchart, sprite, block, command, algorithm, answer, correct, errors, program, algorithm, instructions, commands, forward (fd), left (lt), right (rt), move, turn, clear screen (cs), variable.

UKS2: Coding & Programming

Children build on their programming skills by using new systems such as a flowchart. They continue to break down problems and create algorithms to solve them. They are able to explain the outcome of an algorithm with confidence and accuracy.

KS2 Computing National Curriculum

Children design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; they solve problems by decomposing them into smaller parts. They use sequence, selection, and repetition in programs and work with variables and various forms of input and output. They use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.

Children can:

- a use external triggers and infinite loops to demonstrate control;
- b follow a sequence of instructions, e.g. in a flowchart and modify a flowchart using symbols;
- c use conditional statements and edit variables;
- d decompose a problem into smaller parts to design an algorithm for a specific outcome and use this to write a program;
- e keep testing a program and recognise when it needs to be debugged;
- f use key vocabulary to demonstrate knowledge and understanding in this strand: flowchart, algorithm, control, output, symbol, start, stop, delay, process, decision, loop, backdrop, script, block, repeat, commentary, sequence, consequence, debug, program, Kodu, world, object, tool palette, program environment, smooth, flatten, raise.