Cares Curriculum: Computing

Programming Progression Map Year 1

National Curriculum

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

Y1 Moving a Robot

Programme/Device: eArl coding robot

Vocabulary

Forwards, backwards, turn, clear, go, **commands**, instructions, directions, left, right, plan, **algorithm**, **program**, route,

Knowledge

- What a given command does
- List which commands can be used on a given device
- A program is a set of commands that a computer can run
- a series of instructions can be issued before they are enacted

Skills

- Enact a given word
- Predict the outcome of a command on a device
- Match a command to an outcome
- Run a command on a floor robot
- Choose a command for a given purpose
- Choose a series of words that can be enacted as a program
- Choose a series of commands that can be run as a program
- Build a sequence of commands in steps
- Combine commands in a program
- Run a program on a device

Y1 Programming Animations

Programme/Device: Scratch Jr

Vocabulary

ScratchJr, Bee-Bot, sprite, compare, programming, programming area, block, joining, **command**, start block, **run**, **program**, background, delete, reset, **algorithm**, predict, effect, change, value, instructions, sprite, delete, design,

Knowledge

- words that can be enacted
- commands can be used on a given device
- what a given command does
- how to run a command (press a button)
- a program is a set of commands a computer can run
- a series of instructions can be issued before they are enacted
-

Skills

- enact a given word
- predict the outcome of a command on a device
- match a command to an outcome
- choose a command for a given purpose
- choose a series of words that can be enacted as a program
- choose a series of commands that can be run as a program
- build a sequence of commands in steps
- combine commands in a program
- run a program on a device

Programming Progression Map Year 2

National Curriculum

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

Y2 Robot Algorithms

Programme/Device: eArl coding robot

Vocabulary

Instruction, sequence, clear, unambiguous, **algorithm**, **program**, sequence, order, prediction, design, route, mat, debugging, **decomposition**

Skills

- Choose a series of instructions that can be run as a program
- Create a program
- Trace a sequence to make a prediction
- Run a program on a device
- Debug the program that I have written

Knowledge

- Robots do not make decisions themselves
- A program needs to be started
- A series of instructions is a sequence
- What happens when we change the order of instructions
- A sequence shows the order in which things happen
- When you follow a sequence of instructions, there is an outcome

Recall that a series of instructions can be issued before they are enacted

Y2 Programming Quizzes

Programme/Device: Scratch JR

Vocabulary

Sequence, **command**, **program**, **run**, start, outcome, predict, blocks, sprite, **algorithm**, design, actions, project, modify, change, build, match, compare, **debug**, features, evaluate

Skills

- choose a series of words that can be enacted as a sequence
- explain what happens when we change the order of instructions
- choose a series of commands that can be run as a program
- trace a sequence to make a prediction
- test a prediction by running the sequence
- create and debug a program that I have written
- run a program on a device

Knowledge

- describe a series of instructions as a 'sequence'
- series of instructions can be issued before they are enacted
- use logical reasoning to predict the outcome of a program

Programming Progression Map Year 3

National Curriculum

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

Y3 Sequencing Sounds

Programme/Device: Scratch

Vocabulary

Scratch, programming, blocks, **commands**, **code**, sprite, costume, stage, backdrop, motion, turn, point in direction, go to, glide, sequence, event, task, design, run the code, order, note, chord, design, **algorithm**, bug, **debug**

Skills

- Build a sequence of commands
- Combine commands in a program
- Order commands in a program
- Create a sequence of commands to produce a given output

Knowledge

- Programs start because of an input
- What a sequence is
- A program includes sequences of commands
- The sequence of a program is a process
- The order of commands can affect a program's output
- Different sequences can achieve the same output
- Different sequences can achieve different outputs

Y3 Events and Actions in Programs

Programme/Device: Scratch

Vocabulary

Motion, event, sprite, **algorithm**, logic Move, resize, extension block, pen up, set up, pen, design, event, action, **debugging**, errors, setup, design, code, setup, test, **debug**, actions, events

Skills

- build a sequence of commands
- combine commands in a program
- order commands in a program
- create a sequence of commands to produce a given outcome

Knowledge

- programs start because of an input
- what a sequence is
- a program includes sequences of commands
- the sequence of a program is a process
- the order of commands can affect a program's output
- different sequences can achieve the same output
- different sequences can achieve different outputs

Programming Progression Map Year 4

National Curriculum

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

Y4 Repetition in Shapes

Programme/Device: FMS Logo

Vocabulary

Program, turtle, **commands**, **code** snippet, **algorithm**, design, **debug**, logo commands, pattern, repeat, **repetition**, **count-controlled loop**, value, trace, value, **decompose**, procedure

Skills

- read code and predict outcome
- identify everyday tasks that include repetition as part of a sequence (dance, brushing teeth)
- identify patterns in a sequence
- identify a loop in a sequence
- use an indefinite loop and count controlled loop to produce an outcome
- create two or more sequences that run at the same time
- decompose an every day task (break it down)
- use procedures to create patterns
- design, write and debug a program that includes loops

Knowledge

- typing commands can program a computer
- basic Logo commands
- an algorithm is an ordered set of precise instructions
- repeat means to do or say something again
- a loop command in a program will repeat instructions
- an indefinite loop will run until a program is stopped
- a count controlled loop will stop after a specific numbers of times
- explain the importance of instruction order in a loop
- A procedure is a named code snippet that can be run multiple times.
- in programming there are indefinite loops and count-controlled loops

Y4 Repetition in Games

Programme/Device: Scratch

Vocabulary

Scratch, programming, sprite, blocks, **code**, **loop**, repeat, value, forever, **infinite loop, count-controlled loop**, costume, **repetition**, animate, event block, duplicate, modify, design, sprite, **algorithm**, **debug**, refine, evaluate

Skills

- list an everyday task as a set of instructions including repetition
- identify patterns in a sequence
- identify a loop within a program
- identify patterns in a sequence, eg 'step 3 times' means the same as 'step, step, step'
- use an indefinite loop to produce a given outcome
- use a count-controlled loop to produce a given outcome
- plan a program that includes appropriate loops to produce a given outcome
- create two or more sequences that run at the same time

Knowledge

- what 'repeat' means
- identify everyday tasks that include repetition as part of a sequence, eg brushing teeth, dance moves
- we can use a loop command in a program to repeat instructions
- in programming there are indefinite loops and count-controlled loops
- an indefinite loop will run until the program is stopped
- you can program a loop to stop after a specific number of times
- when to use a loop and when not to
- the importance of instruction order in a loop
- tools enable more than one process to be run at the same time (concurrency)
- not all tools enable more than one process to be run at once

Programming Progression Map Year 5

National Curriculum

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

Y5 Selection in Physical Computing

Programme/Device: Crumble

Vocabulary

Microcontroller, components, connection, **infinite loop**, **output,** motor, **repetition**, **count-controlled loop**, Crumble controller, switch, LED, Sparkle, crocodile clips, connect, battery box, **program**, **Input**, **output**, **selection**, **condition**, action, debug

Skills

- choose a condition to use in a program
- compare a count controlled loop with a condition-controlled loop
- create a condition-controlled loop
- use a condition in an 'if... then…' statement to start an action
- use selection to switch program flow
- use 'if... then... else…' to switch program flow in one of two ways

Knowledge

- a condition can only be true or false
- a count-controlled loop contains a condition
- a condition-controlled loop will stop when a condition is met
- when a condition is met a loop will complete a cycle before it stops
- selection can be used to branch the flow of a program
- a loop can be used to repeatedly check whether a condition has been met
- the importance of instruction order in 'if... then... else...' statements

Y5 Selection in Quizzes

Programme/Device: Scratch

Vocabulary

Selection, **condition**, true, false, **count-controlled loop**, outcomes, conditional statement (the linking together of a condition and outcomes), **algorithm**, **program**, **debug**, question, answer, task, design, **input**, implement, test, **run**, setup

Skills

- choose a condition to use in a program
- compare a count controlled loop with a condition-controlled loop
- create a condition-controlled loop
- use a condition in an 'if... then…' statement to start an action
- use selection to switch program flow
- use 'if... then... else…' to switch program flow in one of two ways

Knowledge

- a condition can only be true or false
- a count-controlled loop contains a condition
- a condition-controlled loop will stop when a condition is met
- when a condition is met a loop will complete a cycle before it stops
- selection can be used to branch the flow of a program
- a loop can be used to repeatedly check whether a condition has been met
- the importance of instruction order in 'if... then... else...' statements

Programming Progression Map UKS2

National Curriculum

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

Y6 Variables in Games

Programme/Device: Scratch

Vocabulary

Variable, change, name, value, set, design, event, improve, evaluate, share, **algorithm**, **code**, task, **program**, project, **code**, test, **debug**

Skills

- identify examples of information that is variable, e.g. a football score during a match
- identify a variable in an existing program
- experiment with the value of an existing variable
- choose a name that identifies the role of a variable to make it more usable (to humans)
- decide where in a program to set a variable
- update a variable with a user input
- use an event in a program to update a variable

Knowledge

- 'variable' is something that is changeable
- a variable can be used in a program, e.g. 'score'
- define a program variable as a placeholder in memory for a single value
- a variable has a name (which must be unique) and a value
- the value of a variable can be used by a program
- variables can hold numbers (integers) or letters (strings)
- a variable can be set as a constant (fixed value)
- the importance of setting up a variable at the start of a program
- there is only one value for a variable at any one time
- if you change the value of a variable, you cannot access the previous value (cannot undo)
- if you read a variable, the value remains

Y6 Sensing

Programme/Device: Scratch

Vocabulary

Micro:bit, MakeCode, **input**, **process**, **output**, flashing, USB, trace, **selection**, **condition**, if then else, **variable**, random, sensing, accelerometer, value, compass, direction, navigation, design, task, **algorithm**, plan, create, **code**, test, **debug**

Skills

- identify examples of information that is variable, e.g. a football score during a match
- identify a variable in an existing program
- experiment with the value of an existing variable
- choose a name that identifies the role of a variable to make it more usable (to humans)
- decide where in a program to set a variable
- update a variable with a user input
- use an event in a program to update a variable

Knowledge

- 'variable' is something that is changeable
- a variable can be used in a program, e.g. 'score'
- define a program variable as a placeholder in memory for a single value
- a variable has a name (which must be unique) and a value
- the value of a variable can be used by a program
- variables can hold numbers (integers) or letters (strings)
- a variable can be set as a constant (fixed value)
- the importance of setting up a variable at the start of a program
- there is only one value for a variable at any one time
- if you change the value of a variable, you cannot access the previous value (cannot undo)
- if you read a variable, the value remains

• use the same variable in more than one location in a program	• the name of a variable is meaningless to the computer	• use the same variable in more than one location in a program	• the name of a variable is meaningless to the computer