The Beeches Primary School – Computing Curriculum Progression Map

Redeveloped – March 2022

Reception Term 1 (Me and my community, Sparkle and shine), Term 2 (Build it up and marvellous machines, Once upon a time puppets and pop ups), Term 3 (Ready, steady grow!, Creep, crawl, wiggle, splash)	
NC – Programmes of Study	To know the name of some machines such as cars, computers, mobile phones etc. To be able to name some different machines and what they are used for such as telephone, computer, washing machine, traffic lights, television, car, bus. To that machines have changed over time by improving with technology and that in the past some machines did not exist.
Knowledge	To identify and understand what a computer is. To identify and understand what a mobile phone is. To identify and understand what a tablet is. To know that scientists began to develop small computers in the second half of the 20th century. To know how technology has changed over the last 50 years.
Vocabulary	Computers, mobile phones, laptop, tablet, telephone, technology, machine, internet, printer, engine, games.

Year 1 On The Move	Learn that programs execute by following clear instructions. Understand that programs respond to inputs to do different things.			
NC – Programmes of Study	Lesson 1 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL)	Lesson 2 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and is able to express simple linear (non-	Lesson 3 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL)	Lesson 4 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL)

		branching) algorithms symbolically. (AL)		
Knowledge	To understand that when a computer does something, it is following instructions called 'code'.	To give instructions to make objects on the screen move when the program starts.	To use code to make objects move when they are clicked on.	To use code to write a computer program where objects in a space scene move when they are clicked on.
Vocabulary	code, instructions, run, up, down, left, right, direction, object, action	instructions, code, action, object	click event, code, action, object, click	program, programming, code, action, object, click, decompose
Progression	In Reception children can name some different machines and what they are used for such as telephone, computer etc. In Year 1 this knowledge is further developed by children understanding when a computer does something, it is following instructions called 'code'.			

Year 1 Simple Inputs	Learn to combine start and input events to create more advanced apps and programs using precise instructions.				
NC - Programme of Study	Lesson 5 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL)	Lesson 6 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL)	Lesson 7 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL)	Lesson 8 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL)	Lesson 9 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL)
Knowledge	To combine start events and click events to make a simple game.	To combine start events and click events to make a simple game.	To combine start events and click events in code to make a magic castle scene.	To combine start events and click events to program cars and emergency vehicles in an animated traffic scene.	To Debug simple inputs. Use coding knowledge to fix the mistakes in a variety of programmes.

Vocabulary	code, object, action, click, start event, click event	click, start, click event, start event, code, object, action	object, action, click, start, code	object, action, click, start, stop, code	click, start, click event, start event, code, object, action
Progression	In Year 1 children will combine start events and click events to make a simple game. This knowledge is further developed in Year 2 as children will write codes which will make an object change direction when different keys on the keyboard are pressed.				

Year 2 Different sorts of inputs	Learn that programs respond to different sorts of inputs, and that the keyboard can be used to control objects on screen, not just by clicking them directly.				
NC - Programme of Study	Lesson 1 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL)	Lesson 2 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL)	Lesson 3 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL)	Lesson 4 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL)	Lesson 5 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL)
Knowledge	To write code that makes an object move around the screen	To make objects perform different actions when keys are	To write code that makes an object change direction when different	To write code that makes an object change direction when the pointer is pressed and released.	To write code where different inputs can be used to make objects move and disappear.

	when keys are pressed.	pressed on the keyboard.	keys on the keyboard are pressed.		
Vocabulary	object, key press, control, action, algorithm, input device	key press event, object, action, input, key	run, execute, direction, code, key press, clockwise, anti-clockwise	pointer, pointer press, pointer release, object, input	input, mouse, pointer, object, device
Progression	In Year 2 children write codes which make an object change direction when different keys on the keyboard are pressed. This knowledge is further developed in Year 3 as children create a program which uses sequences to move two different objects on screen.				

Year 2 Buttons and Instructions	Learn that one object can be used to control another object, e.g. writing code so clicking a button gives an instruction to make a lorry move.				
NC - Programme of Study	Lesson 6 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and is able to express simple linear (non-branching)	Lesson 7 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects	Lesson 8 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) • Demonstrates care and precision to avoid errors. (AL) • Detects and corrects errors, i.e.	Lesson 9 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL)	Lesson 10 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) • Demonstrates care and precision to avoid errors. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL)

	algorithms symbolically. (AL)	errors, i.e. debugging in algorithms. (AL)	debugging in algorithms. (AL)		
Knowledge	To write code where buttons can be used to make an object move around the screen.	To write the code for a simple game where buttons are used to move an object around.	To write the code for a simple game where buttons are used to move an object around and cast a magic 'disappearing spell'.	To write code where buttons are used to move a monster around and eat (hide) fruit.	To Debug simple inputs. Use coding knowledge to fix the mistakes in a variety of programmes.
Vocabulary	button, program, direction, run, execute, control, click, algorithm	button, program, run, execute, control, click, algorithm	button, program, run, execute, control, click, algorithm	button, object, run, execute, algorithm, debug	button, program, direction, run, execute, control, click, algorithm, debug
Progression	In Year 2 children write codes which make an object change direction when different keys on the keyboard are pressed. This knowledge is further developed in Year 3 as children create a program which uses sequences to move two different objects on screen.				

Year 3 Sequence and animation	Learn to make things happen in a sequence, creating simple animations and simulations.			
NC – Programme of Study	Lesson 1 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching)	Lesson 2 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL)	Lesson 3 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL)	Lesson 4 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL)

	algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL)			
Knowledge	To write a computer program where different pieces of code execute in a particular sequence.	To create a program that uses sequences for two different objects moving on the screen.	To write code that uses a timer to create a sequence of events.	To write code that uses a timer to create a sequence of traffic lights turning on and off
Vocabulary	sequence, run, before, after, between, execute, algorithm	sequence, order, before, after, between, action, algorithm, execute	timer event, sequence, run, before, after, execute, algorithm, debug	timer event, sequence, before, after, execute, algorithm
Progression	In Year 3 children write code that uses a timer to create a sequence of traffic lights turning on and off. In Year 4 children use a variable to keep track of the score in a game where the score increases, decreases or resets when different conditions are met.			

Year 3 Conditional events (selection)	Learn to code with 'if statements', which select different pieces of code to execute depending on what happens to other objects.				
NC – Programme of Study	Lesson 5 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an	Lesson 6 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an	Lesson 7 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching)	Lesson 8 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e.	Lesson 9 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL)

	algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL)	algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL)	algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL)	debugging in algorithms. (AL)	
Knowledge	To use 'hit events' to program a space maze game in which an object reacts to particular conditions.	To use conditional hit events to control the movement of a car on the screen.	To make a simple game that uses conditional hit events to check if one object has hit another.	To program a simple game where conditional events are used to check whether objects have collided.	To Debug simple inputs. Use coding knowledge to fix the mistakes in a variety of programmes.
Vocabulary	walls, condition, conditional statement, background, hit event	condition, conditional statement, background, direction, hit event	conditional statement, condition, collide, object, hit event	conditional statement, condition, collide, object, hit event, input	conditional statement, condition, collide, object, hit event, input, dubug
Progression	In Year 3 children write codes that uses a timer to create a sequence of traffic lights turning on and off. This knowledge is further developed as in Year 4 children use a variable to keep track of the score in a game where the score increases, decreases or resets when different conditions are met.				

Year 4 Introduction to variables	Learn how computers use variables to count things and keep track of what is going on, then create simple games which use a score variable.					
NC – Programme of Study	Lesson 1 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that	Lesson 2 • Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that	Lesson 3 Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that	Lesson 4 Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that	Lesson 5 Knows that users can develop their own programs and can demonstrate this by creating a simple program in an environment that	Lesson 6 Debugging Use your coding knowledge to fix the mistakes in a variety of programs.

	does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL) • Designs simple algorithms using loops and selection, i.e. if statements. (AL) • Declares and assigns variables. (AB)	does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL) • Designs simple algorithms using loops and selection, i.e. if statements. (AL) • Declares and assigns variables. (AB)	does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL) • Designs simple algorithms using loops and selection, i.e. if statements. (AL) • Declares and assigns variables. (AB)	does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL) • Designs simple algorithms using loops and selection, i.e. if statements. (AL) • Declares and assigns variables. (AB)	does not rely on text. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL) • Designs simple algorithms using loops and selection, i.e. if statements. (AL) • Declares and assigns variables. (AB)	
Knowledge	To understand how a variable can be used to keep track of the score in a game.	To use variables to keep track of the score in a game that uses conditional events.	To use a variable to keep track of the score in a game that uses conditional events.	To learn how to use multiple different variables and to set the value of a variable.	To use a variable to keep track of the score in a game where the score increases, decreases or resets when different conditions are met.	
Vocabulary	variable, score, start, click, time, alert	variable, conditional event, score, time, value, hit event	variable, value, conditional event,	variable, set, change, cost, total, button	variable, score, event, condition, change, set	

			execute, hit event, negative, collide			
Progression	In Year 4 children learn how to use multiple different variables and learn to set the value of a variable. Knowledge is further developed in Year 5 as children learn to set values in code to control the speed of an object.					

Year 4 Introduction to variables	Repetition and loops Learn how computers use repetition and loops to do things over and over again (and again!).			
NC – Programme of Study	Lesson 7 Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL) • Designs simple algorithms using loops and selection, i.e. if statements. (AL) • Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else (AL) • Uses diagrams to express solutions. (AB)	Lesson 8 Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL) • Designs simple algorithms using loops and selection, i.e. if statements. (AL) • Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else (AL) • Uses diagrams to express solutions. (AB)	Lesson 9 Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL) • Designs simple algorithms using loops and selection, i.e. if statements. (AL) • Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else (AL) • Uses diagrams to express solutions. (AB)	Lesson 10 Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL) • Designs simple algorithms using loops and selection, i.e. if statements. (AL) • Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else (AL) • Uses diagrams to express solutions. (AB)
Knowledge	To use a loop to do something repeatedly in a program.	To write code that uses nested loops to create a car-driving program. Designs simple algorithms using loops and selection, i.e. if statements.	To write the code to program a rocket to orbit round the spinning Moon, using the concepts of loops, regular or infinite repetition, and 'if statement' blocks.	To use loops, a variable and if statements to create an animated scene of hot air balloons performing a repeating pattern in the sky.

Vocabulary	repetition, loop, action, efficient	repetition, loop, nesting, action, efficient, repeat	always, object, event, variable, condition, timer, if statement, loop	loop, repetition, variable, direction, if statement
Progression	In Year 4 children learn how to use multiple different variables and to set the value of a variable. Knowledge is further developed in Year 5 as children learn to set values in code to control the speed of an object.			

Year 5	Speed, direction and coordinates Learn how computers use numbers to represent things such as how fast things are moving, and where they are.					
NC – Programme of Study	Lesson 1 Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL) • Designs simple algorithms using loops and selection, i.e. if statements. (AL) • Designs solutions (algorithms) that use repetition and two-	Lesson 2 Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL) • Designs simple algorithms using loops and selection, i.e. if statements. (AL) • Designs solutions (algorithms) that use repetition and two-	Lesson 3 Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL) • Designs simple algorithms using loops and selection, i.e. if statements. (AL) • Designs solutions (algorithms) that use repetition and two-	Lesson 4 Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL) • Designs simple algorithms using loops and selection, i.e. if statements. (AL) • Designs solutions (algorithms) that use repetition and two-	Lesson 5 To design and create a driving game, using conditions in my code, and explain how my program works	Lesson 6 To debug a programme by focusing on the speed, direction and coordinates.

	way selection, i.e. if, then, and else (AL) • Uses diagrams to express solutions. (AB)	way selection, i.e. if, then, and else (AL) • Uses diagrams to express solutions. (AB) • Declares and assigns variables. (AB)	way selection, i.e. if, then, and else (AL) • Uses diagrams to express solutions. (AB) • Declares and assigns variables. (AB)	way selection, i.e. if, then, and else (AL) • Uses diagrams to express solutions. (AB) • Declares and assigns variables. (AB)		
Knowledge	To set values in code to control the speed of an object.	To use object properties (speed, heading and angle) to create a driving simulation.	To create a sailing game where a boat's position on the screen is controlled by making changes to its co-ordinates.	To write code including if statements to make an object rotate, and combine this with conditional events to make a game.	To set friction to affect the speed and movement of a car in a driving simulation.	
Vocabulary	object, action, speed, property, value, accelerate, decelerate, debug	angle, speed, heading, value, iteratively, object properties, simulation	decomposition, angle, co-ordinates, condition, negative numbers, y-axis, x-axis, position	To write code including if statements to make an object rotate, and combine this with conditional events to make a game.	friction, angle, heading, direction, speed, condition, simulation, overlap	
Progression	In Year 5 children use object properties (speed, heading and angle) to create a driving simulation and in Year 6 children write codes that prompts the user to input the value of a variable, and use this to create an interactive block chart.					

Year 6	More complex variables Learn to use variables in more complex ways, and to manipulate inputs to create useful outputs.			
NC – Programme of Study	Lesson 1 • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL)	Lesson 2 • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-	Lesson 3 • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL) •	Lesson 4 • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL) • Designs simple

	• Detects and corrects errors, i.e. debugging in algorithms. (AL) • Designs simple algorithms using loops and selection, i.e. if statements. (AL) • Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else. (AL) • Uses diagrams to express solutions. (AB) • Declares and assigns variables. (AB)	branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL) • Designs simple algorithms using loops and selection, i.e. if statements. (AL) • Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else. (AL) • Uses diagrams to express solutions. (AB) • Declares and assigns variables. (AB) • Uses post-tested loop, e.g. 'until', and a sequence of selection statements in programs, including an if, then and else statement. (AL)	Designs simple algorithms using loops and selection, i.e. if statements. (AL) • Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else. (AL) • Uses diagrams to express solutions. (AB) • Declares and assigns variables. (AB)	algorithms using loops and selection, i.e. if statements. (AL) • Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else. (AL) • Uses diagrams to express solutions. (AB) • Declares and assigns variables. (AB)
Knowledge	To write code that prompts the user to input the value of a variable and use this to create an interactive block chart.	To use my knowledge of variables to make a balloon pop game that gets harder as users score more points.	To write the code for a shopping till using variables to store and calculate values.	variable, discount, calculate, total, percentage
Vocabulary	input, variable, property, background, grid, pixel, block, convert, value, alignment, unit, scale	variable, condition, event, random, loop, if statement	variable, discount, calculate, total, percentage	variable, discount, calculate, total, percentage

Year 6	Object properties Learn more about how computers use property values and parameters to store information about objects.				
NC – Programme of Study	Lesson 1 • Executes, checks and changes	Lesson 2 • Executes, checks and changes	Lesson 3 • Executes, checks and changes programs. (AL) •	Lesson 4 • Executes, checks and changes programs. (AL) •	Lesson 5 • Executes, checks and changes programs. (AL) •

	programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL) • Designs simple algorithms using loops and selection, i.e. if statements. (AL) • Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else. (AL) • Uses diagrams to express solutions. (AB) • Declares and assigns variables. (AB)	programs. (AL) • Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL) • Designs simple algorithms using loops and selection, i.e. if statements. (AL) • Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else. (AL) • Uses diagrams to express solutions. (AB) • Declares and assigns variables. (AB)	Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL) • Designs simple algorithms using loops and selection, i.e. if statements. (AL) • Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else. (AL) • Uses diagrams to express solutions. (AB) • Declares and assigns variables. (AB)	Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL) • Designs simple algorithms using loops and selection, i.e. if statements. (AL) • Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else. (AL) • Uses diagrams to express solutions. (AB) • Declares and assigns variables. (AB)	Understands that programs execute by following precise instructions. (AL) • Understands what an algorithm is and can express simple linear (non-branching) algorithms symbolically. (AL) • Detects and corrects errors, i.e. debugging in algorithms. (AL) • Designs simple algorithms using loops and selection, i.e. if statements. (AL) • Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else. (AL) • Uses diagrams to express solutions. (AB) • Declares and assigns variables. (AB)
Knowledge	To create a game where players stop objects moving by changing their properties.	To write code that detects the properties of an object and passes the value of these properties (or a set of parameters) to	To make a football game that passes the speed and heading of the pointer's movement to a ball on the screen.	To make a game that moves objects around by getting information from events and passing object properties. • To learn how to pass properties from one object to a second in order to make	To create a golf game by writing code that accesses and uses object properties, including passing the value of these properties to other objects (passing a set of parameters).

		other objects, and to use this to create a space game.		the second object move relative to the first.	
Vocabulary	random, numbers, property, parameter, objects, variable, location, events, values	friction, direction, angle, heading, variable, property, object, parameter, x-co-ordinate, y-co-ordinate	friction, heading, direction, angle, speed, variable, value, parameter, simulation	parameter, object, property, variable, heading, value	simulation, decomposition, parameter, condition, variable, co-ordinates, property, value

KS2: Word processing progression

3	Co2/1.6 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.	• I can use index fingers on keyboard home keys (f/j), use left fingers for a/s/ d/f/g, and use right fingers for h/j/k/l • I can edit the style and effect of my text and images to make my document more engaging and eye-catching. For example, borders and shadows. • I can use cut, copy and paste to quickly duplicate and organise text.
4		• I can combine digital images from different sources, objects, and text to make a final piece of a a variety of tasks: posters, documents, eBooks, scripts, leaflets. • Confidently and regularly use text shortcuts such as cut, copy and paste and delete to organise text • Use font sizes appropriately for audience and purpose.\Use spell check and thesaurus including through Siri and other AI technology

5		• I can start to apply other useful effects to my documents such as hyperlinks. • I can import sounds to accompany and enhance the text in my document. • I can organise and reorganise text on screen to suit a purpose
6		• I can confidently choose the best application to demonstrate my learning. • I can format text to suit a purpose. • I can publish my documents online regularly and discuss the audience and purpose of my content.