Medium-Term Planning
Subject: Computing

Term and Year:	Year 1/2 – Spring 1
Teacher:	Miss Defty
Subject:	Computing Y1: Algorithms Unplugged Y2: Algorithms and Debugging
Vocabulary that will be taught:	Y1 – Algorithm, computer, order, specific, instructions, tasks, solution, bug, virtual, assistance, output, programming, sensor, input, devices, artificial intelligence, automatic, motion, decompose, manageable, problem, decomposition, organising, chunks, code, directions, de-bug, correct Y2 – Algorithm, decomposition, artificial intelligence, data, loops, abstraction, zoomed in, unnecessary, key features, debugging, error, bugs, correcting

National Curriculum Objectives:

- 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

Computing Skills that will be taught and assessed:

Year 1:

- Learning how to explore and tinker with hardware to find out how it works
- Understanding that computers and devices around us use inputs and outputs, identifying some of these
- Learning that decomposition means breaking a problem down into smaller parts
- Using decomposition to solve unplugged challenges
- Using logical reasoning to predict the behaviour of simple programs
- Developing the skills associated with sequencing in unplugged activities
- Learning that an algorithm is a set of step-by-step instructions used to carry out a task, in a specific order
- Follow a basic set of instructions
- Assembling instructions into a simple algorithm
- Learning to debug instructions when things go wrong
- Learning to debug an algorithm in an unplugged scenario

Year 2:

- Articulating what decomposition is
- Decomposing a game to predict the algorithms used to create it
- Using decomposition to decompose a story into smaller parts
- Learning what abstraction is
- Learning that there are different levels of abstraction
- Explaining what an algorithm is
- Following an algorithm
- Creating clear and precise algorithm
- Learning that computers use algorithms to make predictions
- Learning that programs execute by following precise instructions
- Incorporating loops within algorithms
- Using logical thinking to explore software, predicting, testing and explaining what it does
- Using an algorithm to write a basic computer program
- Learning what loops are
- Incorporating loops to make code more efficient

	Focus of each lesson 'Can I…' Statement(s)	Activities/Key points
Lesson 1	**Y1: To understand what an algorithm is** • I can explain that an algorithm is a set of instructions to carry out a task • I know that these instructions need to be carried out in a specific order • I know that computers use algorithms to help them carry out tasks correctly • I can show that there can be more than one solution to solve a problem	Discuss how an algorithm is a clear set of instructions used to carry out a task. Children to practice writing and following algorithms correctly.
	Y2: To decompose a game to predict the algorithms that are used • I understand the definitions: decomposition and algorithm • I can decompose a game to predict algorithms • I can plan algorithms for a more complex game	Discuss how an algorithm is a clear set of instructions used to carry out a task. Children to play a real-life version of the 'Dinosaur move' game, with the pupils being the dinosaur from the game.
Lesson 2	**Y1: To follow instructions precisely to carry out an action** • I can explain why an algorithm must be clear and precise • I can explain the problems a robot can have following our instructions	Children to follow algorithms to draw figures and creatures. Children should be able to consider what helped make to their algorithms more effective, explain why algorithms must be precise and explain why the order of instructions is important.
	Y2: To understand that computers can use algorithms to make predictions • I can explain what an algorithm is • I can explain that computers use algorithms to make predictions • I can write a clear and precise algorithm	Place pupils into pairs and give each pair a number of Lego blocks and explain that they have one minute to build a small model. Children to write a set of instructions so that someone else would be able to recreate their model if they followed the instructions. Ask the children to adapt their algorithms to be even more specific, for example, to specify brick colour and size.
Lesson 3	**Y1: To understand that computers and devices around us use inputs and outputs** • I can identify some input devices • I can identify some output devices • I can identify some devices that are both input and output devices	Explaining to children that inputs are a way of getting information into a computer and that outputs are a way of getting information out of a computer. Also, make it clear that some devices can be input and output devices. Explain that the children are going to be virtual assistants and must do what they are programmed to do by you. As a class, come up with a name for your virtual assistant. The children must respond when you speak to them. For example, let's call the virtual assistant 'Geoffrey': Hey Geoffrey, when I clap, I want you to jump.
	Y2: To plan algorithms that will solve problems • I can devise and create algorithms to solve problems • I can include loops in my algorithms (count controlled) • I can visualise directions from a 2D environment	Pupils to log into a computing device and share the 'Google – Coding for carrots' link with them. Pupils work through the 'Google – Coding for carrots' activities at their own pace.
Lesson 4	**Y1: To understand and be able to explain what decomposition is** • I can explain that decomposition is where you break a problem into small manageable chunks • I understand how decomposition allows you to solve a problem more easily • I can explain how we use decomposition in our everyday lives	Model the process of designing and decomposing before the children work independently. Give each pupil a copy of the Activity: Decomposition comic strip sheet. Ask the children to break down their design. Draw it in step-by-step stages, so that someone else could recreate it. Remind the children that the more precise and detailed, the better. They may want to include labels, arrows or numbers.
	Y2: To understand what abstraction is • I can explain what abstraction is • I can give an example of when abstraction might be useful	Put the children into small groups and give each group photographs of classrooms and key places around your school that you have photographed (hall, playground, field, different classrooms). Explain that they are going to use abstraction to make a plan view of each location to create a collaborative map of the school. Model this activity by drawing a plan of your classroom using simple shapes and limited detail. Discuss what key features should be included so that it's still recognisable.
Lesson 5	**Y1: To know how to debug an algorithm** • I can spot bugs in algorithms • I can fix the error (debug it) and explain the problem it caused	The children are going to use decomposition and debugging skills to solve problems with directions (algorithms) intended to guide them between two areas of a map. Unfortunately, the person who wrote the directions wasn't very good at reading maps and there are lots of 'bugs' (mistakes) in the directions (algorithms).
	Y2: To understand what debugging is • I understand the meaning of the word 'debugging' • I can listen to my peer's verbal instructions • I can perform a task by following step-by-step instructions	. . Give yourself and the pupils five or six Lego bricks of the same size and type. Explain that you are going to make an object out of the pieces and they have to copy it exactly. In silence and with no instructions, model creating the object in front of the children and get them to copy it. The children are going to follow their partner's verbal instructions to build a Lego figure (robot). The job of Child B (Computer) is to build their robot without seeing the model and only listening to Child A's (Programmer) instructions.