Computing Curriculum

Subject Leader: Alistair Scott

Rode Heath Primary School

Contents

Intent

At Rode Heath Primary School there are high expectations for all learners in computing. The computing curriculum encompasses three distinct aspects: computer science (how computers work), information technology (how IT is used) and digital literacy (how IT is used safely and effectively). The core of the computing curriculum is computer science, in which children are taught the principles of information and computation, how digital systems work and how to put this knowledge to use through programming.

Building on this knowledge and understanding, children are equipped to use information technology to create programs, systems and a range of content.

Computing also ensures that children become digitally literate. They are able to use a range of devices to express themselves and develop their ideas through information and communication technology - at a level suitable for the future workplace and as active participants in a digital world.

Therefore, computing enables children to be aware of: knowledge, skills and cultural capital in order to thrive and develop in a digital world. Furthermore, children are encouraged to make links to other subjects which create an interconnected curriculum. In order to achieve this, long-term curriculum plans, and a wide variety of resources are in place to ensure the effectiveness and accuracy of this subject area. Finally, the school's values of: teamwork, resilience, fairness and curiosity are embedded in all that we do.

The national curriculum for computing aims to ensure that all children:

- can understand and apply fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation;
- can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems;
- can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems;
- are responsible, competent, confident and creative users of information and communication technology.

Throughout the computing curriculum at Rode Heath Primary School, children are encouraged to be the best they can be, achieving high academic standards, pushing themselves to deepen their knowledge and skill set. The expectations of children across all year groups is that they should maintain a high standard of work that is the same as the core subjects. We passionately believe that all children can reach these high standards. Pupil premium is used to ensure all children fully access the computing curriculum. We ensure up-to-date hardware and software is available to all. Inclusion is paramount; children with SEN are supported in various ways to suit their needs in order for every child to receive their full curriculum entitlement. This varies from child to child, often by support and the adaptation of resources, including digital devices. We ensure every child is challenged to fulfil their potential.

Implementation

At Rode Heath Primary School we strongly believe that prior knowledge is very important. The scheme of work ensures that all lessons build on prior learning and there are many planned opportunities for analysis and evaluation of recall. This is very important in ensuring progression across the curriculum and Key Stages. Declarative and procedural knowledge is developed throughout the scheme of work. Declarative knowledge is 'knowing that'. It consists of facts, rules, and principles and the relationships between them. For example, features of unreliable content. In contrast, procedural knowledge is 'knowing how'. It is knowledge of methods or processes that can be performed. For example, implementing a repeat in a programming language.

The scheme of work is continually analysed and revised to ensure that a breadth and depth of content is explored as technology changes rapidly. Declarative knowledge ('knowing that') and procedural knowledge ('knowing how') are embedded in the scheme of work. The scheme includes various resources such as: Teach Computing, BBC Bitesize, Lego WeDo, Micro:bit, eAWARE, Raspberry Pi, Scratch and Redfern Electronics.

Lessons aim to engage, inspire and challenge all pupils ensuring that all learners:

- Understand and use key vocabulary.
- Use computational thinking and creativity to solve various problems in an ever-changing way.
- Understand computer science and how computers work.
- Are equipped to express themselves via technology appropriately, so learners are active members in the digital world and future workplaces.

Learning across the school is based on the following key strands:

- Using technology responsibly and safely.
- Programming skills.
- Physical computing skills.
- Digital literacy skills, computer science and information technology.

The importance of and passion for computing should permeate throughout the school. Children should be fascinated by the creativity and benefits of computing. Learning about computing ensures that children will be well-prepared as technology continues to develop:

"Machines take me by surprise with great frequency." Alan Turing

Impact

Learning, understanding and application is monitored via numerous strategies including: formative and summative techniques. Also, practical tasks such as: programming a component or machine to function, demonstrate learners' understanding of a certain topic or problem. End Points are included to show where children should be at the end of their year group. The scheme of work builds on prior learning too, to ensure children are able to progress through the units effectively.

Curriculum Overview Key Stage 1

Year 1	**Digital Literacy/Computer Science**	**Programming (Information Technology, Digital Literacy and Computer Science)**	**Physical Computing (Information Technology, Digital Literacy and Computer Science)**
	Technology in the classroom. Using technology. Developing mouse skills. Using and developing keyboard skills. Data and information – grouping data. *Online Safety lessons are part of the Hearts and Minds Curriculum and are taught each half-term. **Information Technology skills are modelled and developed across all subject areas, such as: how to save and retrieve a document in word; how to create a presentation in PowerPoint and how to create charts and analyse data in Excel.**	What is an algorithm and what is a computer program? Learners will become accustomed to the ScratchJr programming environment. They will discover that they can move characters on-screen using commands, and compare ScratchJr to the Bee-Bots used in the previous unit. Learners will discover that blocks can be joined together in ScratchJr. They will use a Start block to run their programs. They will also learn additional skills such as adding backgrounds and deleting sprites. Learners will follow given algorithms to create simple programs. Learners will discover that some blocks in ScratchJr have numbers underneath them. They will learn how to change these values and	How do you program a robot (Bee-Bots)? This unit introduces learners to early programming concepts. Learners will explore using individual commands, both with other learners and as part of a computer program. They will identify what each floor robot command does and use that knowledge to start predicting the outcome of programs. The unit is paced to ensure time is spent on all aspects of programming and builds knowledge in a structured manner. Learners are also introduced to the early stages of program design through the introduction of algorithms.

		identify the effect on a block of changing a value. Learners will be taught how to add and delete sprites in ScratchJr. They will discover that each sprite has its own programming area, and learn how to add programming blocks to give instructions to each of the sprites. Learners will choose appropriate backgrounds and sprites for a 'Space race' project. They will decide how each sprite will move, and create an algorithm based on the blocks available in ScratchJr that reflects this.	
	Composite Task **These are the fundamentals that help with programming and physical computing skills; therefore, they are continually assessed.**	**Composite Task** Learners will use their project designs from the previous lesson to create their projects on-screen in ScratchJr. They will use their project design, including algorithms created in the previous lesson, to make programs for each of their rocket sprites. They will test whether their algorithms are effective when their programs are run.	**Composite Task** Learners plan their routes before they start to write their programs. The activities also introduce the concept of there being more than one way to solve a problem. This concept applies to a lot of programming activities: the same outcome can be achieved through a number of different approaches, and there isn't necessarily a 'right' way. The lesson also introduces the idea of program design, in which learners need to plan what they want their program to achieve before they start programming.

Year 2 *(Review of Prior Learning)*	**Digital Literacy/Computer Science** What is information technology? Where have we seen information technology in the school? Where have we seen information technology in the world? How does IT improve our world? Data and information - Pictograms *Online Safety lessons are part of the Hearts and Minds Curriculum and are taught each half-term. **Information Technology skills are modelled and developed across all subject areas, such as: how to save and retrieve a document in word; how to create a presentation in PowerPoint and how to create charts and analyse data in Excel.**	**Programming (Information Technology, Digital Literacy and Computer Science)** Recap what an algorithm is. Learners will recap what they know already about the ScratchJr app. They will begin to identify the start of sequences in real-world scenarios, and learn that sequences need to be started in ScratchJr. Learners will create programs and run them in full-screen mode using the Green flag. Learners will discover that a sequence of commands has an 'outcome'. They will predict the outcomes of real-life scenarios and a range of small programs in ScratchJr. Learners will then match programs that produce the same outcome when run, and use a set of blocks to create programs that produce different outcomes when run. Learners will be taught how to use the Start on tap and Go to page (Change background) blocks. They will use a predefined design to create an animation based on the seasons. Learners will then be introduced to the task for the next lesson. They will predict what a given algorithm might mean. Learners will look at an existing quiz design and think about how this can be realised within	**Physical Computing (Information Technology, Digital Literacy and Computer Science)** Build a duck challenge emphasising creativity and problem solving. Lego WeDo projects: Milo the Science Rover: In this lesson, you will: • Explore different ways in which scientists and engineers can research remote places. • Create and program Milo the Science Rover. • Document how Milo can help you to discover a special plant specimen. Milo's motion sensor: In this lesson, you will: • Create and program Milo's object-detector arm using the Motion Sensor Input. • Document how Milo has found the special plant specimen. Milo's tilt sensor: In this lesson, you will: • Create and program Milo's messaging arm using the Tilt Sensor. • Document Milo's communication with the base.

		the ScratchJr app. They will choose backgrounds and characters for their own quiz projects. Learners will modify a given design sheet and create their own quiz questions in ScratchJr.	Collaborating: In this lesson, you will: • Create a document a device to move the plant sample. • Document and present a summary of Milo's mission.
	Composite Task **These are the fundamentals that help with programming and physical computing skills; therefore, they are continually assessed.**	**Composite Task** Learners will create their own quiz question designs including their own choices of question, artwork, and algorithms. They will increase the number of blocks used within their sequences to create more complex programs. Learners will compare their projects to their designs. They will think about how they could improve their designs by adding additional features. They will modify their designs and implement the changes on their devices. Learners will find and correct errors in programs (debug) and discuss whether they debugged errors in their own projects.	**Composite Task** Frog's Metamorphosis. Links to Science curriculum (animals, including humans). In this composite task, you will: • Explore what you know about the stages in the life cycle of a frog, from birth to adult. • Create and program a model of a young frog and an adult frog. • Document the changing characteristics of your model throughout different stages of a frog's life.

Curriculum Overview Key Stage 2

Year 3 *(Review of Prior Learning)*	Digital Literacy/Computer Science	Programming (Information Technology, Digital Literacy and Computer Science)	Physical Computing (Information Technology, Digital Literacy and Computer Science)
	What is Computer Science? How does a digital device work? What parts make up a digital device? How am I connected? How are computers connected? What does our school network look like? Creating a branching database *Online Safety lessons are part of the Hearts and Minds Curriculum and are taught each half-term. **Information Technology skills are modelled and developed across all subject areas, such as: how to save and retrieve a document in word; how to create a presentation in PowerPoint and how to create charts and analyse data in Excel.**	Recap what is an algorithm is. Learners to a new programming environment: Scratch. Learners will begin by comparing Scratch to other programming environments they may have experienced, before familiarising themselves with the basic layout of the screen. Learners will create movement for more than one sprite. In doing this, they will design and implement their code, and then will create code to replicate a given outcome. Finally, they will experiment with new motion blocks. Learners will be introduced to the concept of sequences by joining blocks of code together. They will also learn how event blocks can be used to start a project in a variety of different ways. In doing this, they will apply principles of design to plan and create a project. Learners explore sequences, and how they are implemented in a simple program. Learners have the opportunity to experiment with sequences where order is and is not important. They will create their own sequences from given designs.	Getting started with Crumble: In this beginner Crumble project, you will learn how to set up a Crumble controller, connect common physical computing components such as motors and LEDs to it, and write simple code to make each component work. Police Lights: Create a program to simulate the lights on a police car. Children will add another component (motor) to their program. They will alter the time, speed and direction of the motor. They will program different sequences of the sparkle and the motor so that they can come on together or at different times.

		Learners develop an understanding of sequences by giving them the opportunity to combine motion and sounds in one sequence. They will also learn how to use costumes to change the appearance of a sprite, and backdrops to change the appearance of the stage. They will apply the skills in Activity 1 and 2 to design and create their own project, including sequences, sprites with costumes, and multiple backdrops.	
	Composite Task **These are the fundamentals that help with programming and physical computing skills; therefore, they are continually assessed.**	**Composite Task** Learners will create a musical instrument in Scratch. They will apply the concept of design to help develop programs and use programming blocks — which they have been introduced to throughout the unit. They will learn that code can be copied from one sprite to another, and that projects should be tested to see if they perform as expected. Optional: Summative assessment: Rubric	**Composite Task** Crumble Sparkle Creature: Create a model creature that has Crumble Sparkles as its eyes, and write code to control how the creature's eyes light up.

Year 4 *(Review of Prior Learning)*	**Digital Literacy/Computer Science** Connecting networks What is the internet made of? Sharing information What is a website? Who owns the web? Data – Answering questions *Online Safety lessons are part of the Hearts and Minds Curriculum and are taught each half-term. **Information Technology skills are modelled and developed across all subject areas, such as: how to save and retrieve a document in word; how to create a presentation in PowerPoint and how to create charts and analyse data in Excel.**	**Programming (Information Technology, Digital Literacy and Computer Science)** Learners look at real-life examples of repetition, and identify which parts of instructions are repeated. Learners then use Scratch, a block-based programming environment, to create shapes using count-controlled loops. They consider what the different values in each loop signify, then use existing code to modify and create new code, and work on reading code and predicting what the output will be once the code is run. Learners look at different types of loops: infinite loops and count-controlled loops. They practise using these within Scratch and think about which might be more suitable for different purposes. Learners create designs for an animation of the letters in their names. The animation uses repetition to change the costume (appearance) of the sprite. The letter sprites will all animate together when the event block (green flag) is clicked. When they have designed their animations, the learners will program them in Scratch. After programming, learners then evaluate their work, considering how effectively they used repetition in their code. Learners look at an existing game and match parts of the game with the design. They make	**Physical Computing (Information Technology, Digital Literacy and Computer Science)** In this unit, learners will use physical computing to explore the concept of selection in programming through the use of the Crumble programming environment. Learners will be introduced to a microcontroller (Crumble controller) and learn how to connect and program components (including output devices- LEDs and motors) through the application of their existing programming knowledge. Learners are introduced to conditions as a means of controlling the flow of actions and make use of their knowledge of repetition and conditions when introduced to the concept of selection (through the if, then structure).

		changes to a sprite in the existing game to match the design. They then look at a completed design, and implement the remaining changes in the Scratch game. They add a sprite, and re-use and modify code blocks within loops, and explain the changes made.	
	Composite Task **These are the fundamentals that help with programming and physical computing skills; therefore, they are continually assessed.**	**Composite Task** Learners look at a model project using repetition. They then design their own game based on the model project, producing a design and algorithm for sprites in the game. They share these designs with a partner and have time to make any changes to their design as required. Learners build their games, using the designs they created in lesson 11. They follow their algorithms, fix mistakes and refine designs in their work as they build. They evaluate their work once it is completed, and showcase games at the end. Optional: Summative assessment: Rubric	**Composite Task** Learners will build on the designs they have developed by creating an algorithm to meet the requirements of the given task. They will identify how they are going to use selection before writing their algorithm. They will then move into the code level to test their algorithm by implementing it as a program, running it, identifying any bugs and returning to the algorithm to debug it where necessary. Finally, they will conclude the unit by evaluating their algorithms and other areas of their designs.

Year 5	**Digital Literacy/Computer Science**	**Programming (Information Technology, Digital Literacy and Computer Science)**	**Physical Computing (Information Technology, Digital Literacy and Computer Science)**
	Systems Computer systems and us Searching the web Selecting search results How search results are ranked How are searches influenced Data and Information – Flat-file Databases *Online Safety lessons are part of the Hearts and Minds Curriculum and are taught each half-term. **Information Technology skills are modelled and developed across all subject areas, such as: how to save and retrieve a document in word; how to create a presentation in PowerPoint and how to create charts and analyse data in Excel.**	Learners revisit previous learning on 'selection' and identify how 'conditions' are used to control the flow of actions in a program. They are introduced to the blocks for using conditions in programs using the Scratch programming environment. They modify the conditions in an existing program and identify the impact this has. Learners will develop their understanding of selection by using the 'if… then… else...' structure in algorithms and programs. They will revisit the need to use repetition in selection to ensure that conditions are repeatedly checked. They identify the two outcomes in given programs and how the condition informs which outcome will be selected. Learners use this knowledge to write their own programs that use selection with two outcomes. Learners consider how the 'if… then... else…' structure can be used to identify two responses to a binary question (one with a 'yes or no' answer). They identify that the answer to the question is the 'condition', and use algorithms with a branching structure to	Name badge: Students create their first programs and transfer them to their micro:bits. Beating heart: Create a simple animation to learn about sequence and simple loops. Emotion badge: Start learning about inputs and outputs using buttons and icons on the display. Step counter: Introduce variables to track your step count and begin to use the accelerometer input. Nightlight: Make an automatic nightlight and discover how logic, conditionals and inputs and outputs combine to make a simple control system.

		represent the actions that will be carried out if the condition is true or false. They learn how questions can be asked in Scratch, and how the answer, supplied by the user, is used in the condition to control the outcomes. They use an algorithm to design a program that uses selection to direct the flow of the program based on the answer provided. They implement their algorithm as a program and test whether both outcomes can be achieved. Learners will be provided with a task: to use selection to control the outcomes in an interactive quiz. They will outline the requirements of the task and use an algorithm to show how they will use selection in the quiz to control the outcomes based on the answer given. Learners will complete their designs by using storyboards to identify the questions that will be asked, and the outcomes for both correct and incorrect answers. To demonstrate their understanding of how they are using selection to control the flow of the program, learners will identify which outcomes will be selected based on given responses.	
	Composite Task **These are the fundamentals that help with programming and physical computing skills; therefore, they are continually assessed.**	**Composite Task** Learners will use the Scratch programming environment to implement the first section of their algorithm as a program. They will run the first section of their program to test whether they have correctly used selection to control the outcomes, and debug their program if required. They will then continue implementing their algorithm as a program. Once completed, they will consider the value of sharing their	**Composite Task** Rock, paper, scissors: Combine skills from the previous lessons to turn your micro:bit into an electronic simulation of a popular game of chance.

		program with others so that they can receive feedback. Learners conclude the lesson by using another learner's quiz and providing feedback on it. Learners will return to their completed programs and identify ways in which the program can be improved. They will focus on issues where answers similar to those in the condition are given as inputs, and identify ways to avoid such problems. Learners will also consider how the outcomes may change the program for subsequent users, and identify how they can make use of setup to provide all users with the same experience. They will implement their identified improvements by returning to the Scratch programming environment and adding to their programs. They conclude the unit by identifying how they met the requirements of the given task, and identifying the aspects of the program that worked well, those they improved, and areas that could improve further. Optional: Summative assessment: Selection in quizzes	

Year 6	**Digital Literacy/Computer Science** Internet addresses Data Packets Making your own website Data and information – Spreadsheets *Online Safety lessons are part of the Hearts and Minds Curriculum and are taught each half-term. **Information Technology skills are modelled and developed across all subject areas, such as: how to save and retrieve a document in word; how to create a presentation in PowerPoint and how to create charts and analyse data in Excel.**	**Programming (Information Technology, Digital Literacy and Computer Science)** Learners are introduced to variables. They see examples of real-world variables (score and time in a football match) before they explore them in a Scratch project. Learners then design and make their own project that includes variables. Finally, learners identify that variables are named and that they can be letters (strings) as well as numbers. Learners understand that variables are used in programs, and that they can only hold a single value at a time. They complete an unplugged task that demonstrates the process of changing variables. Then, learners explore why it is important to name variables and apply their learning in a Scratch project in which they make, name, and update variables. Learners apply the concept of variables to enhance an existing game in Scratch. They predict the outcome of changing the same change score block in different parts of a program, then they test their predictions in Scratch. Learners also experiment with using different values in variables, and with using a variable elsewhere in a program. Finally, they add comments to their project to explain how they have met the objectives of the lesson. Learners work at the 'design' level of abstraction, where they create their artwork and algorithms. Learners first design the sprites and backgrounds for their project,	**Physical Computing (Information Technology, Digital Literacy and Computer Science)** What is a Raspberry Pi? What are input and output devices? Use the micro:bit's radio function to send data from one micro:bit to another when the accelerometer detects a shake gesture. Record my voice using the micro:bit's microphone and play it back sped up or slowed down. Use the micro:bit as a data logger. Display an image or text on the micro:bit using the IPO method. Use a 'variable' to track whether an LED should be turned on or off when a push switch is pressed.

		then they design their algorithms to create their program flow. Using debugging skills in order to create a successful algorithm for a drone. 3D Printing – using Tinkercad to design and create CAM mechanisms (DT/Engineering project)	
	Composite Task **These are the fundamentals that help with programming and physical computing skills; therefore, they are continually assessed.**	**Composite Task** Learners implement the algorithms that they created in Lesson 7. In doing this, they identify variables in an unfamiliar project and learn the importance of naming variables. They also have the opportunity to add another variable to enhance their project. Learners build on the project that they created in Lesson 8. They consider how they could improve their own projects and make small changes to achieve this. Learners then have the opportunity to add a variable independently. Finally, learners evaluate each other's projects; they identify features that they liked and features that could be improved. Or Plan first: Use paper to draw a storyboard (unplugged Computing). Create your own game using Scratch or the drone and the Makeblock – Airblock progam on everything you have learnt in KS2. Success criteria: include a	**Composite Task** 2 pages on what the children have learnt in this unit of work. 1 page = theory e.g. IPO and what an algorithm is. 2nd page = what projects we did and how they progressed e.g. using Scratch to then using Python. Evaluate too. And/Or Christmas lights (Links to science – electricity – circuits) Add more LEDs to your breadboard, connect them to your Pi and code a spectacular light show! Peer assess each other's work as a carousel. How will you take what you have learnt in Computing throughout KS1 and KS2 into KS3 and High School (make links to other subject areas e.g. perseverance in PE and debugging in English, editing etc.)

		sprite, backdrop, a sprite that talks and the use of arrow keys, lifting and landing in a specific spot. Peer assess each other's work as a carousel. Optional: You can carry out the summative assessments based on: internet addresses and data packets and introduction to spreadsheets for Year 6, if you so wish to.	

Progression of Skills – Programming KS1

Year Group	Skills from the National Curriculum
1	• Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions • Use technology purposefully to create, organise, store, manipulate and retrieve digital content • Recognise common uses of information technology beyond school • Use logical reasoning to predict the behaviour of simple programs • 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
2	• Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions • 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 • Create and debug simple programs

Progression of Skills – Programming KS2

Year Group	Skills from the National Curriculum
3	• 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 technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact
4	• 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 technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact

	• Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts • 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 sequence, selection, and repetition in programs; work with variables and various forms of input and output
5	• 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 technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact • Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content
6	• 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 • Use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact • 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

Progression of Skills – Physical KS1

Year Group	Skills from the National Curriculum
1	• 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 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
2	• 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 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

Progression of Skills – Physical KS2

Year Group	Skills from the National Curriculum
3	• 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 • 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 logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs • Use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact
4	• 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

	• Use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact. • 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
5	• 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 • 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
6	• 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 • 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 • Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content • 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

Computing End Point Measures	
Year Group	**End Points**
1	**Physical Computing (Information Technology, Digital Literacy and Computer Science** • Explain what a given command will do. • Act out a given word. • Combine forwards and backwards commands to make a sequence. • Combine four direction commands to make sequences. • Plan a simple program. • Assess and evaluate my knowledge and understanding of this unit of work. **Programming Computing (Information Technology, Digital Literacy and Computer Science)** • Explain how technology helps us. • Name the parts of a computer and use a mouse to click and drag. • Use a mouse to create a picture. • Type my name and save my work. • Use more keys on the keyboard. • Describe objects and put them into groups. • Understand what an algorithm is. • Choose a command for a given purpose. • Show that a series of commands can be joined together. • Identify the effect of changing a value. • Explain that each sprite has its own instructions. • Design the parts of a project. • Assess and evaluate my knowledge and understanding of this unit of work.

2	**<u>Physical Computing (Information Technology, Digital Literacy and Computer Science</u>** • Be creative when producing a computer program. • Program Milo using the Motion Sensor Input. • Program Milo using the Tilt Sensor. • Program Milo to move an object. • Assess and evaluate my knowledge and understanding of this unit of work. **<u>Programming Computing (Information Technology, Digital Literacy and Computer Science)</u>** • Describe some uses of computers. • Identify information technology in the school. • Identify information technology beyond the school. • Explain how information technology benefits us. • Understand what data is and how it can be presented. • Explain that a sequence of commands has a start. • Explain that a sequence of commands has an outcome. • Create a program using a given design. • Change a given design. • Create a program using my own design. • Assess and evaluate my knowledge and understanding of this unit of work.
3	**<u>Physical Computing (Information Technology, Digital Literacy and Computer Science</u>** • Set up the Crumble controller and connect and code physical components to it. • Use repetition to control physical outputs. • Use an input to control an output. • Assess and evaluate my knowledge and understanding of this unit of work. **<u>Programming Computing (Information Technology, Digital Literacy and Computer Science)</u>** • Understand the concepts of input, process and output for all digital devices. • Design a device that uses an input, process and output.

	• Explain how a computer network can be used to share information. • Explain the role of a switch, server, and wireless access point in a network. • Recognise the physical components of a network. • Create a branching database. • Explore a new programming environment. • Identify that commands have an outcome. • Explain that a program has a start. • Recognise that a sequence of commands can have an order. • Change the appearance of my project. • Assess and evaluate my knowledge and understanding of this unit of work.
4	**Physical Computing (Information Technology, Digital Literacy and Computer Science** • Program a microcontroller to make an LED switch on. • Write a program that includes count-controlled loops. • Explain that a loop can stop when a condition is met. • Explain that a loop can be used to repeatedly check whether a condition has been met. • Design a physical project that includes selection. • Assess and evaluate my knowledge and understanding of this unit of work. **Programming Computing (Information Technology, Digital Literacy and Computer Science)** • Describe how networks physically connect to other networks and why they need protecting. • Recognise how network devices make up the internet and the World Wide Web is part of this. • Describe where websites are stored and the types of media that can be shared. • Describe how content can be added and accessed on the World Wide Web. • Explain that websites and their content are created by people and there are rules to protect this. • Explain that data gathered over time can be used to answer questions.

	• Develop the use of count-controlled loops in a different programming environment. • Explain that in programming there are infinite loops and count controlled loops. • Develop a design that includes two or more loops which run at the same time. • Modify an infinite loop in a given program. • Design a project that includes repetition. • Assess and evaluate my knowledge and understanding of this unit of work.
5	**Physical Computing (Information Technology, Digital Literacy and Computer Science** • Give the micro:bit instructions (an algorithm) in code to make a name badge using the LED display output? • Create a micro:bit animation using a sequence of images in a loop? • Make the micro:bit show different pictures on the LED display output depending on which button input is pressed? • Turn my micro:bit into a step counter using the accelerometer and variables? • Code a micro:bit to make a light that switches on when it gets dark using sensors and logic? • Assess and evaluate my knowledge and understanding of this unit of work. **Programming Computing (Information Technology, Digital Literacy and Computer Science)** • Explain that computers can be connected together to form systems. • Recognise the role of computer systems in our lives. • Experiment with search engines. • Describe how search engines select results. • Explain how search results are ranked. • Recognise why the order of results is important, and to whom. • Know what a flat-file database is and how to use it effectively. • Explain how selection is used in computer programs. • Relate that a conditional statement connects a condition to an outcome. • Explain how selection directs the flow of a program. • Design a program which uses selection. • Create a program which uses selection.

	• Assess and evaluate my knowledge and understanding of this unit of work.
6	**Physical Computing (Information Technology, Digital Literacy and Computer Science** • Understand what the I (Input) P (Process) O (Output) model is. • Use electronic components and an algorithm to create a Scratch game. • Add a switch to a simple circuit. • Use Python to control an LED and button. • Assess and evaluate my knowledge and understanding of this unit of work. **Programming Computing (Information Technology, Digital Literacy and Computer Science)** • Explain the importance of internet addresses. • Recognise how data is transferred across the internet. • Code my own website using HTML (HyperText Markup Language). • Use formulas to calculate PE data and present it using graphs and charts. • Define a 'variable' as something that is changeable. • Explain why a variable is used in a program. • Choose how to improve a game by using variables. • Design a project that builds on a given example. • Use my design to create a project. • Use debugging skills in order to create a successful algorithm for a drone. • Understand and use cam mechanisms in toys to make them move. • Assess and evaluate my knowledge and understanding of this unit of work.