<table>
<tr><th><b>Year Group:</b></th><th>3</th><th>Strand: 2a</th><th>Programming</th></tr>
<tr><td colspan="4"><b>How can I program music using Scratch?</b></td></tr>
<tr><td><b>Key NC Objectives</b></td><td colspan="3">• design, write and debug programs that accomplish specific goals,, solve problems by decomposing them into smaller parts
• use sequence, selection, and repetition in programs
• 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</td></tr>
<tr><td><b>Unit Objectives</b></td><td colspan="3">• Explore a new programming environment (Scratch Online)
• identify the objects in a Scratch project (sprites, backdrops) and recognise that commands in Scratch are represented as blocks
• Identify that each sprite is controlled by the commands I choose.
• Explain that a program has a start and can be started in different ways.
• Create a sequence of connected commands, explaining what a sequence is and that it needs to have an order.
• Create a program to move a sprite in four directions.
• Change the appearance of my project by adding in multiple sprites and deciding the actions for each of them.
• Create a project from a task description or following a design and starting to be able to do this with increasing independence.</td></tr>
<tr><td><b>Suggested learning activities</b></td><td colspan="3">• Build on Knowledge from iPad Scratch Jnr at KS1, exploring Scratch on the laptops. Explore new programming layout, identify the stage, script writing area, sprites and backdrops. Give learners the opportunity to experiment with three motion blocks — move, turn, and point in direction. Explore how sprites can have different costumes to change their appearance and how to add sounds. See Teach Computing materials: https://teachcomputing.org/resources Year 3 Unit -Programming A – Sequences in music – Lesson 1.
• Pupils will create movement for more than one sprite. 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. Pupils will use the blocks, point in direction, turn, Go to (random position), Glide to (random position). See Teach Computing materials: https://teachcomputing.org/resources Year 3 Unit -Programming A – Sequences in music – Lesson 2.
• Pupils learns how to use event blocks and how these help to create sequences. Pupils create sequences to make three different sprites move. See Teach Computing materials: https://teachcomputing.org/resources Year 3 Unit -Programming A – Sequences in music – Lesson 3.
• Explore sound blocks and how to start playing a sound. Pupils order sounds and create their own sequences. See Teach Computing materials: https://teachcomputing.org/resources Year 3 Unit -Programming A – Sequences in music – Lesson 4.
• Pupils explore how to change the appearance of a sprite and the background and become confident in the use of costumes. Children understand the difference between a sprite and its costumes and between coding a sprite and putting code on the background. See Teach Computing materials: https://teachcomputing.org/resources Year 3 Unit -Programming A – Sequences in music – Lesson 5.
• Pupils create their own program and sequence to play a musical instrument using the blocks they have learn over this unit. They apply the learning to a final project. This could be published on class blogs and children gain feedback and evaluations from each other. See Teach Computing materials: https://teachcomputing.org/resources Year 3 Unit -Programming A – Sequences in music – Lesson 6.</td></tr>
</table>

Extra Links and Planning Resources	See links in Core Substantive Knowledge. Extra resources available at: https://scratch.mit.edu/ Lesson support available at: https://teachcomputing.org/	
Online Safety	**3.5: Age appropriate activity** Pupils will understand that not everything on the internet is appropriate for children. See Sheffield Online Safety Curriculum LKS2 Lifestyle and Health L3 for more information.	
Previously Taught Vocabulary	Algorithm, Program, Sequence, Blocks, Sprites, Commands, Debug	
New Key Vocabulary	**Backdrops:** The picture **Costume:** used to change the appearance of a sprite. **Event Blocks:** these blocks cause an event to start a script linked to a particular sprite. **Motion:** movement	**Programming Blocks:** The blocks used to program the sprites. **Script Writing:** The area used to place blocks to create sequences and programs. **Stage:** The background of a project, performs functions through scripting.
Core Substantive Knowledge	Teachers need to have a good understanding of the pedagogy behind teaching computer programming and computational thinking. They need to have a good understanding of how the software Scratch works. This unit explores the concept of sequencing in programming through Scratch. It begins with an introduction to the programming environment, which will be new to most learners. They will be introduced to a selection of motion, sound, and event blocks which they will use to create their own programs, featuring sequences. The final project is to make a representation of a piano. The unit is paced to focus on all aspects of sequences, and make sure that knowledge is built in a structured manner. Learners also apply stages of program design through this unit. (Teach Computing) Key areas of Scratch for explanations: **Blocks palette:** The blocks build on those used in ScratchJr, with several additional functions. Highlight that there are more types of blocks, and explain that this lesson will focus on the motion blocks. **Code area:** This is where blocks are placed to create a program. Learners will do this for themselves following this slide. **Stage with sprite:** The output of the program is presented on the stage. By default the sprite is 'Scratch the cat' and the backdrop of the stage is blank. Learners will change the backdrop later in this lesson. **Run the code:** Learners run the code, using the events (e.g. click the sprite, pressing the keyboard key, or the green flag) included in their design and project. If their project does not work as expected, they may need to debug their design and/or code. There is also a focus on developing pupil's understanding of how to create designs in programming through four levels (abstraction – see below in computational thinking). Research suggests that this structure can support learners in understanding how to create a program and how it works: • Task - what is needed • Design - what it should do • Code - how it is done • Running the code - what it does Spending time at the task and design levels before engaging in code-writing can aid learners in assessing the 'do-ability' of their programs. It also reduces a learner's cognitive load during programming. (Teach Computing)	

	Developing your subject knowledge of Scratch: **Teacher subject knowledge development online courses at:** https://teachcomputing.org/primary-teachers **Scratch online courses:** https://www.futurelearn.com/courses/teaching-programming-primary-school **Computational Thinking:** Computational thinking is about looking at a problem in a way in which a computer can help us to solve it. This is a two-step process: 1. First, we think about the steps needed to solve a problem. 2. Then, we use our technical skills to get the computer working on the problem. Computational Thinking is all about solving problems effectively - with or without a computer Computational thinking involves six different concepts and five approaches to working. Computation Concepts Logic Algorithms Decomposition Patterns Abstraction Evaluation Computational Approaches: Tinkering Creating Debugging Persevering Collaborating For more information on computational thinking: **Computational thinking and concepts:** https://www.barefootcomputing.org/concept-approaches/computational-thinking-concepts-and-approaches **Introduction to Computational Thinking:** https://www.bbc.co.uk/bitesize/guides/zp92mp3/revision/1 **Guidance on Computational Thinking:** https://community.computingatschool.org.uk/resources/2324/single
Prior Knowledge	In previous units, pupils will have used Bee Bots and Scratch Jnr. They will have joined coding blocks together to create algorithms. They will have learnt how to debug a program they have created. See units in Y2.
Assessment	Pupils that are secure in this unit can: Pupils will have explored the concept of sequencing in programming through Scratch. They will have had an introduction to the programming environment and how this is more advanced than Scratch Jnr. Pupils will be introduced to a selection of motion, sound and event blocks which they

	will use to create their own programs, including sequences. Towards the end of the unit, the children will apply the skills they have learnt to create a program for a musical instrument. This will be assessed through outcomes, observations and questioning in lessons.