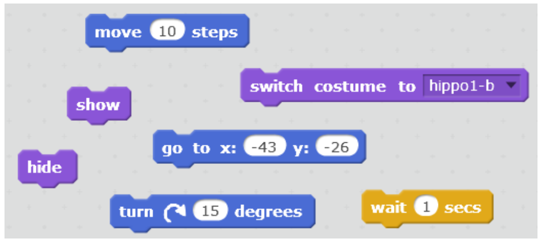
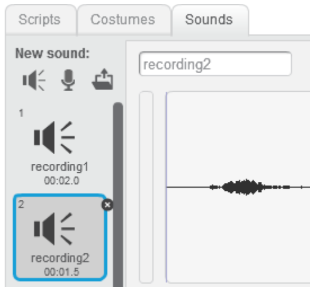
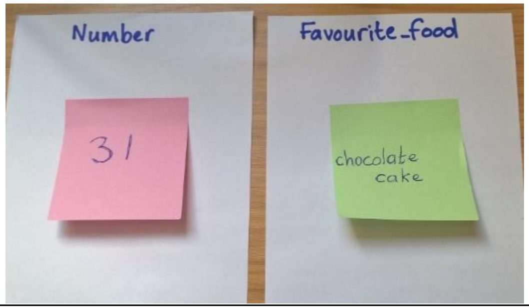
Year 5 Programming 1 Scratch My Roman Numerals (Core) inc RAG and PRIMM

Timing 6 sessions (approximately 45 minutes each)	**Children will** • Review their knowledge of Scratch and share what they can do • Learn about variables and how to use them in a program • Make a variable to count. • Be part of making an animation of an Ancient Roman counting in Roman Numerals. • Make an animation of a modern-day person and Ancient Roman counting together. • Record their voices to add to the counting animation they created
e-safety links I can talk about the dangers of spending too long online or playing a game	**Objectives** **Programming** • I can use a variable to increase programming possibilities. • I can use 'if' and 'then' commands to select an action. • I can decompose a problem into smaller parts to design an algorithm for a specific outcome and use this to write a program. • I can use logical reasoning to detect and debug mistakes in a program. • I can change an input to a program to achieve a different output.
Links to other learning **Maths:** understanding of Roman Numerals; counting in number sequence.	
Resources Scratch 3 (or Scratch 2.0 download if working on Windows 7 or Mac devices) Post it notes – two colours A5 bits of paper Presentation slides	**Preparation** • Download Scratch Desktop from https://scratch.mit.edu/download or use Scratch online linking with Technology in our lives 3 • Download presentation slides to support sessions • Have a real bucket with a label saying count ready for session 2 • For session 4, look at example of a design that has been 'Ragged' • For session 3, see examples simple design, hexagonal design. that could be used to model the process with a class • Session 5 has links to examples of what the programs can look like. • Ask technician to set up links to these projects or download and save on your school drive ○ Counting in session 2 https://scratch.mit.edu/projects/323465474/editor/ ○ Ancient Roman counting in session 3 https://scratch.mit.edu/projects/323481031/editor/ ○ Blank template project in session 4 https://scratch.mit.edu/projects/195990069/editor/ ○ PRIMM project to use with less confident learners in session 4 https://scratch.mit.edu/projects/323468805/editor/

Year 5 Programming 1 Scratch My Roman Numerals (Core) inc RAG and PRIMM

	Expectations	Activity	Success Criteria
1	**Programming** I can change an input to a program to achieve a different output.	**Review of Scratch and recording a sound file** The time taken on this session will vary dependant on how secure the learners are in their knowledge and previous use of Scratch. Go straight to 7 block challenge if you know class are confident users. • Ask children what they may already know about Scratch from previous learning. • **Low confidence:** ○ Add and edit background / sprites ○ Focus on groups of blocks in turn: What can you make happen with individual blocks (direct drive)? Put into short sequences. Work through blue movement blocks, purple looks blocks (limit to top eight blocks initially), yellow control wait block. ○ ***Avoid using Events blocks at this point.*** *It is a common misconception that you must have one of these for a program to run. Click on individual blocks or the top block in a sequence to run a program.* ○ Add in yellow control Repetition (repeat a sequence) and Forever (keep doing the actions) blocks. • **7 block challenge:** ○ Slide 2: What can the children make happen? Stick to the rules on the slide. *(you may want to add 'if on edge bounce' block if learners are having difficulty with a disappearing sprite!)*  move 10 steps switch costume to hippo1-b show go to x: -43 y: -26 hide turn 15 degrees wait 1 secs ○ Let children show outcomes to each other. Use this time to observe confidence of children. How creative have they been using the blocks? What level of knowledge do they show in using the blocks? Check they realise what the 'go to x, y' block does.	Gold: Can I imaginatively create an engaging sequence? Silver: Can I create an engaging sequence? Bronze: Can I put blocks into a sequence to make something happen?

		• **Check knowledge of (pink) sound blocks:** ○ Direct drive - click on sound blocks individually, what does it do? ○ Challenge children to add a sound to their sprite and create more than one sprite with different sounds attached to them. Children will need to click on the Sounds tab for the sprite and can choose the microphone to record their own sound. ○ With a second sprite children will need to use an event block to make both sequences begin at the same time. Direct children to use'when green flag is clicked' or 'when space is pressed'. Check they realised the purpose of the event block here is to start both sequences together.  Scripts Costumes Sounds New sound: recording2 1 recording1 00:02.0 2 recording2 00:01.5 • **Think, pair, share:** What have you been able to do? What have you found out about the software? Have you added a sound to your sprite?	Gold: Can I create my own sound file for a sprite? Silver: Can I choose more than one sound and link this to a sprite? Bronze: Can I link a sprite to a sound I have chosen?
2	**Programming** I can use a variable to increase programming possibilities.	**Introducing variables** • **Paired chat:** (Slide 1) What is a variable? What do we understand by the word variable? What could a variable be? • **Whole class:** Take ideas from class and 'assess' how many may have used variables before, how many have inferred a meaning for the word variable. All should have linked to the idea that something can vary – it is changeable. • **Direct teaching:** We are going to build our understanding of what a variable is when we are programming. • Slide 3: Give children A5 pieces of paper. One to be titled Number, the other Favourite_food. • 'You have all got two variables. One is Number, the other your Favourite_food.' • Children to have two different coloured post-it notes. Tell children to write a number on one of the colours of post-it  	Gold: Can I tell you what a variable is, create a variable and assign different values to it? Silver: Can I talk about variables and create a variable? Bronze: Can I change the value of a variable?

notes and their favourite food on the other colour. Put the post-it notes on the appropriate piece of paper.

- ‘You have assigned a value to your variables’
- Tell your talking partner the value you have assigned to Number. They must use the sentence: ‘I have assigned a value of X to Number.’
- Tell your talking partner the value you have assigned to Favourite_food. They must use the sentence: ‘I have assigned a value of XXX to Favourite_food.’
- Slides 4 and 5: Give the children sentences that includes the variables. (The slides are animated to show one sentence at a time. They may want to assign a different value as you read out the sentences in slide 4.)
- Once you assign a value to variable it will stay as that value and can be used in programmes. Each time the variable is used the same value will be used until you change the value of the variable.
- You can repeat this activity with other variables. Try using a variable linked to current topic or book the class is reading and let children come up with sentences where they will be using the variable and can change the value of the variable.

- **Predict**
  - Pairs: Slide 6 What will this code make happen?
  - Interrupt quite quickly, ‘What did you decide?’ See if they recognise that ‘set length of line’ is setting a value for that variable. Tell them length of line is a variable. Ask them again to predict what the code will make happen. The length of line is set to 0 so the number of repeats is 0 so nothing will happen.

- **Run**
  - Let children run the code. Give them a link to https://scratch.mit.edu/projects/323465474/editor/ or download project and save on your local drive for children to use.

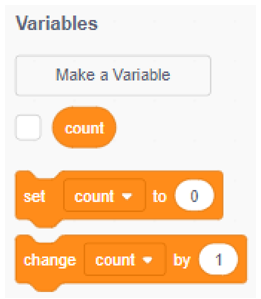
- **Investigate**
  - Tell children to see what happens when they assign a different value to length of line variable.
  - Check children recognise what the erase all (*clear screen*) and go to X Y (*cat goes back to starting position*) blocks do.
  - Review with class: ‘What happens when you change the value for the variable length of Line?’ *They are changing how far the cat moves and therefore how long the line is*.

- **Direct teaching**

A variable stores a value. The value can be a number or a bit of text. It can be an image. We’re going to use a variable to count.

- **Predict**
  - Pairs: Slide 7 What will this code make happen?
  - What is the variable? What happens to the variable?

- **Run**
  - Show children how to create a new variable in Scratch and let them put the sequence together and then run the sequence.



Select Variables (orange blocks)

Select Make a Variable

Give the variable a name (leave as the default ‘For all sprites’ – this means the variable can be used for any sprites in the program.)

Variable blocks can then be used in sequences.

- **Investigate:** Why is repeat 11 times used when the sprite is counting to 10? (*The count starts with a 0 so 11 numbers are shown including the 0.*) What happens if you assign a new value to Count? (*You can count in 1s, 2s, 3s etc.*)

		• **Modify**: Can you make different things happen as the sprite counts? • **Direct teaching** Demonstrate a variable as being like a bucket which will store a number. Have a real bucket with a label saying count. ○ We are going to tell the bucket that, to begin, the count is 1. Put 1 in the bucket (*or use cubes*). ○ We are going to change our variable called count by 1 – we are adding 1. Put another 1 in the bucket. What does count equal now? Repeat until count = 10. ○ (*Choose whether you want to change the numbers for the new number or whether you will keep a pile of 1s in the bucket – this may depend on the understanding of your learners*). • Ask the class to tell each other what a variable is. What else could you use a variable for? eg to create a score. Remind children a variable can be text or an image – not just numbers.	
3	**Programming** (I can decompose a problem into smaller parts to design an algorithm for a specific outcome and use this to write a program.) I use logical thinking, imagination and creativity to extend a program.	**Count in Roman Numerals** • Slide 7: Remind children of the sequence used to Count. What is the algorithm for this sequence? Let children have a go in pairs. Remind children the algorithm is the sequence of things you need to happen. It isn't the blocks of code that you will use to implement it as a program. You could repeat the demonstration of 'counting with a bucket' to provide additional support. • Slide 8: This slide is animated to show one possible algorithm, followed by another to include setting a starting value for count. • What would be different if it was someone counting in Ancient Rome? How would a Roman count? What could this look like? (This assumes the children will have had a maths experience of counting in Roman Numerals – or it could be used as a maths lesson to introduce them.) • Support children to talk through what they would need to do to make a Roman count in Roman numerals. Do they realise they will need to use images for the Roman numerals? They will need to make the image change as the Roman counts. • Model process of designing and making animation of a Roman counting. Use one of these templates: Box planning A4 Hexagon planning A3 Simple planning; or a blank sheet of paper.	Gold: Can I talk through the process of designing and programming an algorithm then use logical thinking and imagination to extend it? Silver: Can I tell you how to design and program an algorithm using a pattern I have used before?

		Model the planning process (Your suggested script is in italics): • **Task:** Model clarifying what the task is. *Need to have an Ancient Roman and show them counting in Roman numerals.* • **Design:** ○ What will it look like? *Draw Ancient Roman with a speech bubble.* ○ What will happen on the screen? *Show the Roman numeral in the speech bubble changing.* • **Think through algorithm** ○ How will it happen? *Roman numeral for one will appear, then two, then three … The algorithm will be:* ▪ *Show Roman numeral for one* ▪ *Keep showing the next Roman numeral* • **Plan the process to achieve this** – *which Scratch commands could be used to make this happen? I will need a Roman numeral sprite. It will need to change for each count. This could be different costumes for one sprite. It could be a different sprite for each count. We'll need to decide which is the easiest way to program this.* *If we use one sprite with different costumes we could use 'Next costume' block to change to the next Roman Numeral. That could be easier than making different sprites appear on the screen.* • **Model** ragging your design and algorithm: ○ Red: 'I do not know which objects, background and blocks to use to make this happen … yet.' ○ Yellow: 'I think I know what to do to make this happen, but I am not sure. I am happy to have a go.' ○ Green: 'I know which objects, background and blocks to use to make this happen.' You can go to town on your acting here! Highlight something Red if you want to get the children to explain it to you. Label something Yellow to model that you are ready to have a go at that bit even though you're not sure. Green to show you are confident that you know how to do that bit – could be adding sprite to project – link to online safety by searching for clipart to use and then	Bronze: Can I make changes to a program?

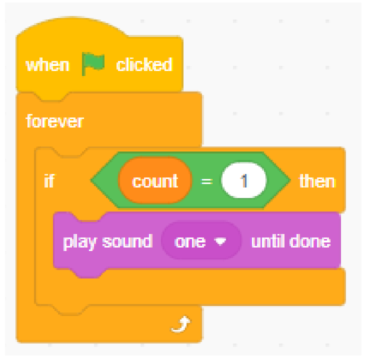
		acknowledging where it came from. The Scratch project examples included in this planning use clipart from http://www.phillipmartin.info/. • **Model** putting the sequence together **OR** show the sequence and get the children to Predict, Run and Investigate the program. https://scratch.mit.edu/projects/323481031/editor/ • **Model** continually checking / debugging, evaluating / improving. Can the children help you to improve the animation? eg *I'd like the animation to start with the Ancient Roman on the screen without the speech bubble.* I could use Show and Hide blocks but I'm not sure how. I need to go back to my algorithm: ▪ *Hide speech bubble and Roman numeral* ▪ *Show speech bubble and Roman numeral for one* ▪ *Keep showing the next Roman numeral in the speech bubble* (You can tidy it up a bit more by hiding speech bubble and Roman numeral when you have finished counting.) • **Use and Modify:** Give children a link to the project or download to your local drive for them to access there. Ask them to run the project and then to modify to make it better using the show and hide blocks.	
4	I can decompose a problem into smaller parts to design an algorithm for a specific outcome and use this to write a program. I use logical thinking, imagination and creativity to	**Make an animation** ***This can be done in pairs or individually. If working individually it is good to agree a talking partner so that there is discussion at each stage about how they will make things work.*** • Slide 9: Set the task to create an animation of a modern-day person and an Ancient Roman counting at the same time. • Provide a template to plan a 'Counting Today and in Roman Times'. You can download one of these: Box planning A4 Hexagon planning A3 Simple planning; or some children may find it easier to use a blank sheet of paper. • Ask the children to follow the planning process you modelled in the last session:	Gold: Can I design and program an algorithm then use logical thinking and imagination to extend it? Silver: Can I design and program an algorithm? Bronze: Can I program an

	extend a program.	○ **Task:** Do I understand what I need to do? Encourage children to think about how they will use what they have learnt so far. ○ **Design:** (Need to draw this out showing that there will be one algorithm for the modern-day person and one for the Ancient Roman) ▪ What will it look like? ▪ What will happen on the screen? Can children label the different parts of their design? What programming would each part need? How will the Roman numerals appear in a Scratch project? One Roman numeral with different costumes? ○ **Think through algorithm** ▪ How will it happen? Do they remember to use a variable for the modern-day person to count? How will they make sure the number 1 appears at the same time as the Roman numeral I and then each number following? ▪ **Plan the process to achieve this** • Prompt them to think about the images they will need to use. • Ask the children to RAG their design / algorithm (see example) as a self-assessment to see what knowledge and skills they already have that they can apply.	algorithm with support?

RAG: Self-assessment / Formative assessment Support Poster

- **RAG** the design/algorithm, (highlighters/underline with coloured pencils).
  - Red: ‘I do not know which objects, background and blocks to use to make this happen … yet.’
  - Yellow: ‘I think I know what to do to make this happen but I am not sure. I am happy to have a go.’
  - Green: ‘I know which objects, background and blocks to use to make this happen.’
- Use the self-assessment to **identify** children that may need:
  - Red: additional direct teaching
  - Yellow: Reinforcement
  - Green: confident to work independently

		• Encourage children to carry out their plan, continually checking / debugging, evaluating / improving. Ask children to tick each part of their algorithm once they have got it working. You can provide this starter project https://scratch.mit.edu/projects/195990069/editor/ so that children have the sprites and costumes they need. They will need to draw a speech bublle sprite if they want to include that. • **To provide support for children that are not confident to have a go. Look at Roman numeral project Predict, Run, Investigate, Modify.** • Ask children to evaluate their animations. Pairs could evaluate each other's using two stars and a wish. • **Class review:** What could be added to improve the animations? This could include a background for modern-day and Ancient Rome. It could include sound. If this is suggested tell children that they will be having a go at this in the next session. Session 6 will provide time for children to make changes to their animation – including adding sound. They will find out how to incorporate sound in session 5.	
5	**Programming** *I can use a sound as an output in Scratch.* **I can use logical reasoning to detect and debug mistakes in a program.** I can decompose a problem into smaller parts to design an algorithm for a	**Count out loud** ***Set up pairs of similar programming confidence for this session. During the session you can differentiate the design and algorithm which would be suitable for the pairs. Examples of solutions are given for you to see what the outcomes could look like. These can be used as part of a PRIMM approach (Predict, run, investigate, modify, make) if children are finding it too difficult to achieve the task. However, a big purpose of this session is to detect and debug mistakes. We need children to be comfortable making mistakes and for them to be independent in sorting them out. An important rule for this session could be, 'You're not allowed to ask an adult to sort out your program.' Of course, you will still intervene where you know particular children are becoming too frustrated, but it is a good principle to stick to.*** • Pairs: What is the algorithm for getting a sprite to count? • Guide them to talk through the algorithm in slide 8 if they are unsure. • Pairs: How would you design an animation of someone counting aloud? How would you change the algorithm to have a voice counting aloud as each number appears? *The algorithm could be the same but it will need to be implemented in a different way. A more precise algorithm is needed*	Gold: Can I detect and debug mistakes while I design an imaginative solution to achieve the outcome I want? Silver: Can I detect and debug mistakes while I design and program an algorithm? Bronze: Can I program an algorithm with support?

	specific outcome and use this to write a program.	• Review algorithms on Slide 10. The slide is animated to talk about the first and then the second design. First design will use idea of selection – if a number appears, play the sound recording of that number. (refer to Year 4 Core Programming unit if children are unsure of concept of selection). Second design has the person counting with the numbers showing and at the same time a sequence of voice recordings playing. • **Emphasise the importance of continually running their program as they put together the blocks.** Running the program as they do each section will help them to check that part and to debug the code where it is necessary. It can be more difficult to sport mistakes if you write the whole program and then try it out. Trying out a bit at a time helps to detect and debug the mistake. Remind children that making mistakes is part of being a programmer. **Mistakes help us to achieve the best solution.** • Pairs: Work on your design and algorithm. How will you implement this as code? You can allow choice of which design and algorithm or assign which each pair should develop. For both designs it will be helpful for children to know that they can right click 'duplicate' to duplicate a sequence and then to change values. • Design 1 will require use of selection concept – if count = X then play recording of X. A sound recording will need to be done for each number and an 'if, then' block used to play the sound when each value of Count appears. The block always needs to be within a forever loop as the program has to check forever if the count is that number. If the forever loop isn't there it will only check once.  when clicked forever if count = 1 then play sound one until done This can be implemented with each if then as a separate sequence that begins with a green flag block. Look at [this example]() if you want to see what this looks like. Or it can be one sequence that uses if… then… else… blocks. Look at [this example]() to see what it looks like. • Design 2 has the counting sequence running and alongside it a sequence of the voice recordings. The children will need to run the sequence of voice recordings to see if any wait blocks are needed to slow it down at any point so that the voice recording of each number coincides with the timing of the changes of the count. It may just need one wait block at the beginning. The number of wait	

		blocks will depend on the timing of each voice recording. Look at this example to see what it looks like. **Challenge for early finishers:** • Can you count backwards? What would the algorithm be? Which programming blocks would you need to change? • Review the outcomes for the pairs with the class. What are you most proud of from your work in this session? How did you sort out mistakes in your programs?	
6	**Programming** I use logical thinking, imagination and creativity to extend a program. I can use logical reasoning to detect and debug mistakes in a program.	**Improve your animation** ***This can be done in pairs or individually. If working individually it is good to agree a talking partner so that there is discussion at each stage about how they will make things work.*** • Slide 11: Remind children of improvements they suggested for their animations at the end of session 4. List the possibilities. These could include, sound, movement, background, changes of background during the counting. • Ask children to make changes to their design and algorithm to incorporate the changes they want to make. Make sure your confident programmers are challenging themselves sufficiently. • Give children time to make changes. Remind them to continually run their program to check it is doing what they want. • Provide an opportunity for children to show their animations to each other. • Agree a set of achievements made by the class during the making of the animations. Make a list of any things they would want to do better the next time they are set a challenge with Scratch.	Gold: Can I use creativity and imagination to improve my work, making changes to my design and using logical thinking to implement those changes? Silver: Can I use creativity and imagination to improve my design and program? Bronze: Can I improve my program?
I use logical thinking, imagination and creativity to extend a program.		**Additional possibilities:** Scratch 3 has possibilities to include additional multimedia learning in the sound activity. Edit Portions of a Sound File: Select and edit portions of a sound. Apply effects (such as faster, slower, etc.) to the selected portions.	

	Copy, Paste, and Delete selected portions of a sound file. You can copy a specific part of a sound and paste it into a different sound file. Copy to New: Select portion of a sound file, use "Copy to new" button to create a new sound file with the portion of the sound you have selected. Effects: include mute, fade in, and fade out. Keyboard Shortcuts that work in the sound editor: ○ Copy = "Ctrl/Command" + "c" ○ Paste = "Ctrl/Command" + "v" ○ Select All = "Ctrl/Command" + "a" ○ Undo = "Ctrl/Command" + "z" ○ Start sound = "Space" ○ Delete = "Backspace/Delete" ○ Delete everything but the selection = "Shift" + "Backspace/Delete