**Computing Medium Term Planning**

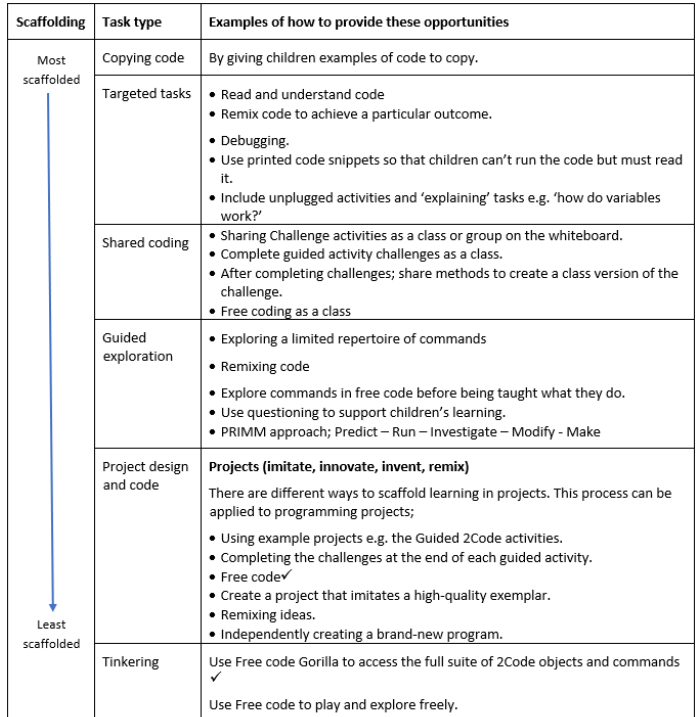
|  |  |  |
| --- | --- | --- |
| **Term:** Summer 2 | **Year:** 3/4 | **Topic/Unit:**3.10 Micro:bit, Coding 4.1 |
| **Key Vocabulary**  **Lesson 1- input, algorithm, code, output, program, hardware, LED, repeat, software**  **Lesson 2**- **animation, image, infinite loop, LED, sequence, data, Output, selection, hardware, repeat, software**  **Lesson 3- animation, image, infinite loop, LED, sequence, data, input, Output, selection**  **Lesson 4- input, algorithm, code, output, accelerometer, gestures, sound output, speaker, ‘if’ statements**  **Lesson 5- selection, ‘if’ statement, command, attributes, execute, repeat until, ‘if/else’ statements, inputs.**  **Lesson 6-** **selection, if’ statement, repeat until, ‘if/else’ statements, inputs, variable, co-ordinate, timer, alert, prompt**  **Lesson 7- selection, if’ statement, repeat until, ‘if/else’ statements, inputs, variable, co-ordinate, timer, alert, prompt** | | |

**Evidence: Please can all class teachers generate an example of work that is Working towards, Working at and Working to GD for each unit and upload them to the computing curriculum folder (or email to Louise) for monitoring. Evidence is collected for Computing Folder in SLT room.**

**Important:**

Follow PRIMM method: Predict… what this code will do, Run… the code to check your prediction, Investigate… trace thought the code to see if you were correct, Modify... the code to add detail, change actions/outcome, Make… a new program that uses the same ideas in a different way.

Adaptions: If this is one of your first coding lessons with the micro:bit, it may be hard to know which children will need more support. Each lesson contains tasks that all children will have a go at completing. Some children will need more peer and/or teacher support to complete these successfully. Working in mixed ability pair groupings may help to ensure all pupils are able to participate in each lesson’s main activities. Each lesson also contains extension ideas to challenge those children who complete the task more quickly than others. The extension activities are optional and may not be completed by all pupils. Please see the Assessment Guidance section at the end of this document to support your teacher judgements on pupil outcomes throughout the lessons.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **National Curriculum** | **Week** | **NC Coverage** | **Skills taught** | **Knowledge** | **Activity Outline** |
| 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 | 1 | 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. | • Understand the micro:bit is a tiny computer which needs instructions in code to make it work.  • Use Free Code micro:bit to create instructions in code that the micro:bit can understand and then transfer them to the micro:bit.  • Know the Micro:bit has an LED display output which it can use to show words (as well as numbers and pictures). | • Children can explain that the micro:bit is a tiny computer.  • Children can give the micro:bit instructions in code to make a name badge using the LED display  output. | *Preparation: set “Free code name badge activity” as 2Do. Select the following objectives:*    Discuss new vocabulary with the children on vocab slide.  Slide 5: recap questions  Slide 6: introduction to micro:bits (some children in year 4 are familiar with them)  Slide 10+: show children how to go through PRIMM process. Model building the code yourself. Run your program using the simulator to test your code before connecting to the micro:bit.  If you need to debug your code (make changes to the code so that it runs as you want it to), do this and run the code again in the simulator. Now that you have built your program and tested it in the simulator, you’re ready to transfer it so that it will run on a real micro:bit.  Watch the video on how to connect a micro:bit to your computer to transfer and run the code or follow the instructions on the next slide. Transfer the code to the micro:bit and use battery packs if wish.  Slide 16: Children open 'Free code micro:bit' which has been set as a 2Do in preparation for the lesson and complete the code (adaption could be made by class teacher for less cognitive load for SEND by doing a section of the steps). Children transfer code to their micro:bit and test.  If you have a battery pack, encourage children to unplug Micro:bits from computers and attach batteries. Their code remains on the Micro:bit and will still work.. Children to test via simulation before transfer to micro:bit. Then evaluate as necessary.  Plenary:  Review meaning of vocabulary (click on words to reveal the definitions) |
| 2 | 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. | • Understand that sequence and timing is  important when making an animation.  • Understand that animations create an illusion  of movement by showing a sequence of still  images.  • Code the micro:bit to show simple animations  on its LED display output. | •Children can create a micro:bit  animation using a sequence of images in a loop.  • I can explain that the order or  sequence of instructions is important. | *Preparation:*  *Set Free Code micro:bit as a 2Do.*    Discuss new vocabulary with the children on vocab slide.  Go through PM slides, recap what they remember about micro:bits.  Slide 6: Watch the video with your class to introduce them to the beating heart animation they will be programming in this lesson. Optionally play the video from YouTube: [https://youtu.be/FpOL7qBbYSg](https://youtu.be/FpOL7qBbYSg%20)  Explain: animation is a sequence of still images shown one after the other to look like movement.  Loops allow us to repeat sets of instructions without having to write them out multiple times.  Ask children: where else you have seen animations? E.g., cartoons, games.  Slide 7: examine and build the code: Read through the code and see if children can *predict* what will happen in the simulator when the program is run.  Explain:  The ‘repeat forever’ block is a loop which keeps the sequence going.  It’s an ‘infinite loop’ – a loop that keeps going as long as the micro:bit has power.The ‘show image’ and ‘sleep’ blocks make up the sequence. Changing the sleep time makes the animation faster or slower.  At the end of the sequence, the loop goes back to the top and starts again. Using a loop means we can use only 4 blocks to create an animation that runs forever.  Slide 8: Watch the video on how to build the beating heart code. You can click on the link to open Free Code Micro:bit and model building the code yourself. Optionally show the class the YouTube coding video: <https://youtu.be/d-beK-lxTaI>Slide 9: run the program simulator to test the code. This is your opportunity to debug (remind children to turn and talk then share what debugging means), then run simulator again if needed.  Slides 10-11- transfer the code to the actual micro:bit. (video to follow if needed)  Activity 1 : Children open Free code micro:bit which has been set as a 2Do in preparation for the lesson and complete the code. Children recreate the code in the Free Code micro:bit, using their own LED s, testing it in the simulator. Children transfer code to their micro:bit and test. If you have a battery pack, encourage children to unplug micro:bits from computers and attach batteries. Their code remains on the micro:bit and will still work.  Ask children to hand work in.  **Share positive examples on the board of childrens work. (Good time to add comment to children’s work for marking- expectation of everyone’s work commented on at least once per half term)**  **Good questioning for evaluations for chn to discuss to aid your marking:**  • Does it work as you expect?  • If not, do you need to debug the code and download it again?  • How good is the project?  • Would you recommend it to a friend? • How could you improve it?  • Could it have other uses?  • How does it work?  • Encourage children to think about how it works when holding it in their hands.  Plenary:  Review meaning of vocabulary (click on words to reveal the definitions) |
| 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.  use logical reasoning to explain how some simple algorithms work and to detect and  correct errors in algorithms and programs. | • Code the micro:bit to make different outputs happen depending on different inputs.  • Understand that inputs and outputs involve the flow of data in and out of computers.  • Apply this knowledge using the micro:bit’s button inputs and display output | • Children can make the micro:bit show different pictures on the LED display output depending  on which button input is pressed.  • Children can explain that inputs are data sent to a computer.  • Children can explain that outputs are data sent from a computer | *Preparation:*  • Free Code micro:bit Emotions link to activity to set as 2Do  • Student handout (optional) – LED planning sheet    Recap vocabulary from previous work, introduce new vocab for lesson. Go through purple Mash slides. (lesson follows same process as previous one; create, test, debug, run and evaluate.  Activity 1: Children open Free code micro:bit which has been set as a 2Do in preparation for the lesson and complete the code. Children recreate the code in Free Code micro:bit, using their own LED emotion images, testing it in the simulator.  Adapted version: You can create a premade version with some standard emotions already installed then encourage the children to adapt the existing code.  **Share positive examples on the board of childrens work. (Good time to add comment to children’s work for marking- expectation of everyone’s work commented on at least once per half term)**  **Good questioning for evaluations for chn to discuss to aid your marking:**  • Does it work as you expect?  • If not, do you need to debug the code and download it again?  • How good is the project?  • Would you recommend it to a friend? • How could you improve it?  • Could it have other uses?  • How does it work?  • Encourage children to think about how it works when holding it in their hands.  Plenary:  Review meaning of vocabulary (click on words to reveal the definitions) |
| 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. | Understand how sensor inputs from the accelerometer can be used to detect movement.  Understand how to create sounds and music using the music editor.  Apply this knowledge using the micro:bit gesture inputs and sound output | • Children can use the music editor to create sounds and music.  • Children can explain that accelerometer is a sensor, an input that senses movement.  • Children can create code that makes sounds play using different movement gestures. | *Preparation:*  *Free Code micro:bit Sounds and Gestures link to activity to set as a 2Do.*  *You can select the following objectives for assessment:*    Recap vocabulary from previous work, introduce new vocab for lesson. Go through purple Mash slides. (lesson follows same process as previous one; create, test, debug, run and evaluate.  Activity 1: Children open Free code micro:bit which has been set as a 2Do in preparation for the lesson and complete the code. Children recreate the code in Free Code micro:bit, using their own LED emotion images, testing it in the simulator.  Adapted version: You can create a premade version with some standard commands already installed then encourage the children to adapt the existing code.  **Share positive examples on the board of childrens work. (Good time to add comment to children’s work for marking- expectation of everyone’s work commented on at least once per half term)**  **Good questioning for evaluations for chn to discuss to aid your marking:**  • Does it work as you expect?  • If not, do you need to debug the code and download it again?  • How good is the project?  • Would you recommend it to a friend? • How could you improve it?  • Could it have other uses?  • How does it work?  • Encourage children to think about how it works when holding it in their hands.  Plenary:  Review meaning of vocabulary (click on words to reveal the definitions) |
| 5  Unit 4.1 coding  Lesson 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.  . | To understand the repeat until command.  To begin to understand selection in computer programming.  To understand how an IF/ ELSE statement works. | • Children can interpret a flowchart that depicts an IF/ ELSE statement.  • Children can read code that includes Repeat Until and IF/ ELSE and explain how it  works.  • Children can create a program that includes an IF/ ELSE statement. | *Preparation:*  *Set Free Code Gibbon as a 2Do.*  *• Set Reginald Rocket 2Code Example as a 2Do. You can select the following objectives when setting the 2Dos to make future assessment easier:*  *• Print copies of the Storyboard Planner for children to use if you are using it (see step 7)*  *Teacher video to watch that aids understanding*  Lesson outline:  Go through PM slides, introducing vocabulary.  IF: start the lesson, identifying with the children how they have used ‘if’ commands previously (using the micro:bits), compare them to the Is it Raining 2Code example (from lesson 2 of unit 4.1). Explain that today we are going to start by looking at how to program something to happen if the condition is not met e.g. program something to happen in our scene if it is not raining. Show example of an if statement that matches the rain scene.    IF/ELSE: Display slide 6. Say to the children ‘IF my class is quiet for 30 seconds, then I will [insert action/ activity here!!] Start a timer and then check IF statement is true. If it is, carry out stated action/ activity. In pairs, ask one child to write an IF statement on their small whiteboard, then the other to check if it’s true and run the action if it is, or not if it isn’t. Discuss as a  class: When tested, were any not true?Explain that in code we can use IF statements to help our programs work – for example, IF the countdown has reached 0 the game is over, or IF the score equals 10 the ‘amazing’ sound plays. Show children the Is It Raining IF ELSE flowchart:    Talk it through with the children and then go back to the Is It Raining 2Code activity and ask them to help you add code to program it (ask them to look at the code blocks and see if they can pick out IF/ ELSE as a sensible one to use, if not direct them to it). The code should look like this:    Run the program a couple of times, once typing ‘yes’ in the prompt for input box, once typing ‘no’. Notice how the code executes and what happens each time.  Reginald Rocket: Display slide 7. Load Reginald Rocket 2Code example. Look at the design first, and then the code. Look at the first part of the code and pick out the command. Can children ‘read’ the code to see what this command is doing? When the user clicks on Reginald (the rocket), a message is printed to the screen – Prepare for Launch – then Reginald will move right (adding 1 to the X attribute) this repeats until the X is greater than the X position of Terry.  Ask children what/who is Terry? - Terry is the launch pad (you can work this out by looking for the object called Terry in Design View). And what is the purpose of this first section of code? - This first section of code moves the rocket onto the launch pad.  Now look at the next part of the code:    What do children think will happen when this program is run? When the children click on Reginald, he will move along to the launchpad. If the input is ‘shoo’ the sheep will run out of the way, if not (else) Reginald will take off with the sheep!  Slide 8. Show children the Reginald If Else Flowchart – if they look on the ‘statement is true’ side Reginald doesn’t take off. What code would they need to add for Reginald to take off 3 seconds after the sheep was shooed away? Can we add a blast off sound?    Activity 1: children tasked to create a a short program that used ‘repeat until’ and ‘if/else’ commands.  **Share positive examples on the board of childrens work. (Good time to add comment to children’s work for marking- expectation of everyone’s work commented on at least once per half term)**  Plenary:  Review meaning of vocabulary (click on words to reveal the definitions) |
| 6 & 7  4.1 lesson 6 across both lessons | 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. | To review vocabulary and concepts learnt in Year 4 Coding.  To create a playable game | • Children can use the correct code to make their game work.  • Children can explain how their code makes their game work. | *Preparation:*  *Set Free Code Gibbon as a 2Do.*  *You can select the following objectives when setting the 2Dos to make future assessment easier:*    *•Print copies of the Storyboard Planner for children to use.*  *• Create a display board for the class to share their programs to. Details of how to do this are given in Appendix 1*  *This lesson is stretched across two weeks.*  Go through PM slides, introducing vocabulary.  Turtle race: Use slide 5. Display Turtle Race game on the board, look at the design with the children.  Click on ‘rs’ and ‘ps’ and look across to the attributes table – notice they are number objects with a value set to 0. These can display a variable value.  Click on Run at the top to run the program and see that they display as 0 at the start.  Click on the stop button and return to Design view. Click on the food, turtles and button and see how they are names in the attributes table. Ask the children to speculate as to how this game might be played.  Slide 6. Exit the design using the ‘See Code’ button and look at the code, give the children a chance to ‘read’ it – you might want to split it into two parts as suggested in slide 6. Give them time with a talking partner to discuss what the code will do.  Relate this to the teaching from the previous lesson on variables – what is going to change the values of the 2 number objects?  Rather than variables, this program uses number objects, when the turtles collide with the food the value attribute of the object changes.  Click on Run to run the code and click on the red and purple race buttons in turn until one turtle has eaten all its food. Which turtle won? The number objects keep a count of how much food the turtles have eaten, like a score.  Give children 2 minutes with their talking partner to discuss how this game could be improved. Ask them to feedback to the class and encourage them to consider if they think these are features, they could add in 2Code.  Free code gibbon: Use slide 9. Open Free Code Gibbon, click on ‘Design’ and add a background and objects, use the attributes table to name them appropriately. Add a number variable called ‘score’. Add a number variable called ‘score’. Your scene might look something like this:  Click on ‘See Code’ and add code that programs one object to collide with another and a score to increase. Add code so that the object collided into jumps to random X/ Y co-ordinates. Ask the children to predict what this will do.  Your code could look something like this:    Ask children to consider how this game might start and finish – how will the user know what to do.  Activity 1: children to use the storyboard planner to design their game and how it will work. They can have Free Code Gibbon open when they do this so they can see the backgrounds and objects that are available.  Activity 2: Once children have finished their designs, they have a go at creating them in Free Code Gibbon. Remind them of the design – code – test – debug process.  These are some of the things they have learned so far – they might want to consider using some of them:  • Number variables to keep score  • Selection – IF and IF/ELSE statements  • Co-ordinates (using X and Y)  • Timers – after and every (could be handy to set a time limit)  • Repeat and Repeat until  • Alert boxes (these could be good for instructions  Get children to save their work to the 2Displayboard and as a class, playa few childrens games and celebrate their achievements.  **Ask children to ‘hand in’ tasks with an honest review of how**  **they got on.**  **Share positive examples on the board of childrens work. (Good time to add comment to children’s work for marking- expectation of everyone’s work commented on at least once per half term)**  Plenary:  Review meaning of vocabulary (click on words to reveal the definitions) |
|  |  |  |  |  |  |