Our Values	Enjoy learning	Try our best	Make good choices	Respect each other & our surroundings	Work together	Celebrate our successes
	learn	try	choices	respect	together	success

SUBJECT MEDIUM TERM PLANNING - COMPUTING

Year Group: 2	**TERM:** Autumn 2	**Theme:** Robot Algorithms

National Curriculum:

- 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 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.

Context: - This unit develops learners' understanding of instructions in sequences and the use of logical reasoning to predict outcomes. Learners will use given commands in different orders to investigate how the order affects the outcome. They will also learn about design in programming. They will develop artwork and test it for use in a program. They will design algorithms and then test those algorithms as programs and debug them.	**Concepts:** Programs Computational Thinking	**Vocabulary:** Order, sequence, program, commands, instructions, algorithm, clear, debug, test

Prior Knowledge: - Be able to explain what a given command will do. - Be able to combine four direction commands to make sequences - Be able to plan a simple program - Be able to find more than one solution to a problem.	**Future Knowledge**: • Design, write and debug programs that can accomplish specific goals, including controlling or stimulating physical systems; solve problems by decomposing them into smaller parts **(KS2).**

- Be able to choose a command for a given purpose. - Be able to show that a series of commands can be joined together. - Be able to identify the effect of changing a value. - Be able to design the parts of a project. - Be able to use own algorithms to create a program.	• Use sequence, selection and repetition in programs; work with variables and various forms of output and input **(KS2).** • Use logical reasoning to explain how simple algorithms work and to detect errors in algorithms and programs **(KS2).**

End points /by the end of this unit pupils will…

- Be able to describe a series of instructions as a sequence
- Be able to explain what happens when we change the order of instructions.
- Be able to follow a sequence
- Be able to predict the outcome of a sequence
- Be able to compare my prediction to the program outcome
- Be able to explain the choices that I made for my mat design
- Be able to identify different routes around a mat
- Be able to test a mat to make sure that it is usable
- Be able to explain what my algorithm should achieve
- Be able to create an algorithm to meet a goal
- Be able to use my algorithm to create a program
- Be able to create and debug a program that I have written

Lesson Number - 1

Key learning: To describe a series of instructions as a sequence and explain what happens when we change the order the order of instructions.

Concepts:
Programs
Computational Thinking

Success Criteria:

- I can follow instructions given by someone else
- I can choose a series of words that can be enacted as a sequence
- I can give clear and unambiguous instructions
- I can create different algorithms for a range of sequences (using the same commands)
- I can use an algorithm to program a sequence on a floor robot
- I can show the difference in outcomes between

Suggested resources:
Flipchart

Whiteboards / paper

Beebots

Floor Mats

Lesson structure: Introduction, direct teaching, activities, key questions

Engage:
Tell pupils that you are going to give them instructions to draw something on their boards or paper. Tell them that they should only draw what they are told, but do not tell them what the object is that they are drawing. Slowly describe to pupils what they should draw on their boards, one part at a time, for example (HOUSE):

- Draw a medium-sized square in the middle of your board
- On the top edge of the square, draw a triangle that is as wide as the square and about half as tall, etc

Note: The instructions that you give to pupils should be carefully pitched. They should provide enough detail for pupils to produce drawings that are similar to the original. This activity is also a model for pupils to follow later.

When you have finished describing the object, ask pupils to show you and each other their drawings. Highlight that the instructions needed to be clear, and that pupils had to listen and follow carefully.

Ask pupils to clear their boards, and give them another set of instructions to draw something else. Repeat the activity as time allows, choosing different drawings from the sheet or using your own ideas.

Note: If time allows, you may wish to choose pupils and ask them to try giving instructions to the class. You may also wish to repeat this activity with the class another time.

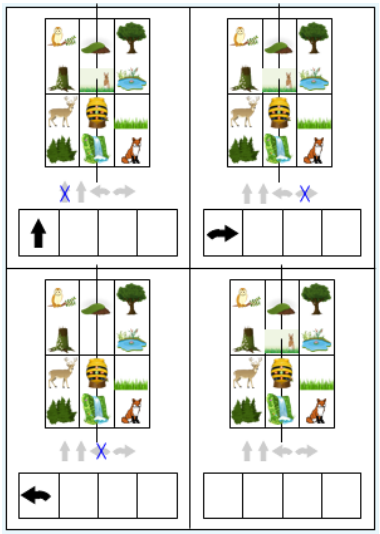
Introduce:
Ask pupils how do we get computers to do what we want. Ask pupils to consider the question and then talk to a partner about their thoughts.

Guide pupils to the idea that we give computers instructions, and instructions may be given by clicking, tapping, or key presses. Pupils may also suggest that we program computers.

Q – when else might we need to use or follow instructions? (At school, following a recipe, playing a game).

P&C:
Tell pupils that they will be giving instructions to a partner who will pretend to be a robot. Explain that

two sequences that consist of the same commands		these instructions need to move the robot (pupil) around the room, and the robot (pupil) cannot speak; they can only follow instructions. Q – is this a clear instruction? "Go forwards" NO – it isn't clear as it doesn't indicate how far. The robot may start moving forwards and continue until they hit a wall. Reinforce to pupils that their directions must be clear, precise, and doable. Ask pupils to spend a few minutes talking to a partner about words or phrases that could be used to give directions and record them on a whiteboard. Some examples of suitable instructions are: move one step forwards, move one step backwards, turn a quarter turn, turn to face something, stop, etc. **1 –** pupils issue and follow one instruction at a time. Remind the pupils that the "robot" cannot speak. After 2 minutes, allow pupils to swap roles. Once both have had a turn, ask the following questions: • Did any of the instructions not work very well? • How could the instructions be improved? **2 -** tell pupils that this time they will give two or three instructions at a time. Explain that the programmer will need to say "Go" to tell the robot to run (follow) the set of instructions, so that the robot knows that the programmer has finished giving their instructions. Tell pupils that the programmer will need to carefully watch their robot to make sure that the robot does as they intended. After 2 minutes, allow pupils to swap roles. ***Scaffolding opportunity –*** *display a list of good instructions around the room that pupils can use.* **Independent:** Show a picture of a floor robot. Explain that floor robots have a computer inside, and they use it to follow instructions. Robots cannot make choices themselves. Q – How do you give a robot instructions?

		Tell pupils you give instructions by pressing buttons. When we give robots or computers an instruction, it is called a **command.** Q – How is it different to giving a partner instructions? Tell pupils that the sets of commands (instructions) that they give to robots or computers can be called 'algorithms'. An algorithm is a precise set of instructions which can be turned into a code by the robot or computer. *CHECKPOINT* *Q – what do you think this means? (loelh) The answer is "hello"* *Q – why is it important that the letters are in the correct order?* *When the letters are in the wrong order, they don't have any meaning. So when we give commands to robots, the order of commands is very important.* Tell pupils that in this activity, they will create different algorithms using four given arrow instructions (forwards, forwards, left turn, right turn). Explain that they should arrange the four arrows in different orders to make four different algorithms. Explain that they need to complete all four sets of boxes. Tell pupils to put the arrows in the boxes, and to cross off each grey arrow when they have used it. Emphasise that the grey arrows are there to limit which commands they can use; this lesson is about using the same set of commands in different orders. *(Activity sheet found in NCCE Computing - Robot Algorithms – Lesson 2)* 

		Deepen: Ask pupils to use the floor robots to try each of their algorithms, one algorithm at a time. Explain that as pupils press the buttons on the robot, they are creating a program. The program is how the algorithm is run as code on the robot. Pupils may need support to systematically enter their programs. Suggest to pupils that they tick, cross off, or cover each command as they enter it into the robot. Suggest that one pupil presses the buttons while another pupil keeps track of the commands as they are entered. **Note:** You may need to remind pupils to press X on the floor robot to clear its memory before they try each program. Tell pupils that when they have finished running a program, they should mark on their activity sheet where the robot got to. Explain that in doing this, pupils are recording the outcome of each algorithm (and program). **Note:** Tell pupils that some programs may take the robot one square off either side of the mat, and that if that happens, they should still record this last position. **Reflection:** Talk through the activity that pupils have done. Q – what were the four commands they used in every algorithm? (forwards, forwards, left, right). Q – did the robot always end up in the same place? Q – did the robot always follow the same route? Highlight that although the same commands were used every time, the order in which they were used changed the outcome of their programs.
Vocabulary: instructions, program, sequence, order, algorithm, commands, clear		

Lesson Number - 2

<table>
<tr><td>Key learning: To use logical reasoning to predict the outcome of a program (series of commands)</td><td>Concepts:
Programs
Computational Thinking</td><td rowspan="2">Lesson structure: Introduction, direct teaching, activities, key questions

Engage:
Q – what is an algorithm? An algorithm is a set of precise instructions which can be turned into a code by a computer / robot.

Q – what kind of instructions can we give to a floor robot?
(move forwards, move backwards, turn right, turn left)

Introduce:
Q – what is a prediction? (A statement about what you think will happen in the future).
Q – how can we make predictions based on instructions? (By thinking through the instructions step-by-step to work out the end outcome).

Show pupils a short series of instructions. (e.g. Sit down, pat your head, stand up). Ask them to think through the sequence shown. Tell pupils that they don't need to move yet. Ask them to predict what would happen if they followed the instructions. After they have made some suggestions, you may wish to ask pupils to follow the instructions physically and see how good their predictions were.

Do the same with a different set of instructions.

CHECKPOINT
Example where the prediction is incorrect. Can pupils identify the mistake?

P&C:
Pupils will follow some given instructions and predict where the robot will stop. Highlight the importance of starting with the robot on the correct square). Model an example on the board using the paper-bot to follow a set of instructions on the large mats from the previous lesson to make their prediction. Describe each direction instruction as you model.

Pupils will need the 'Follow the algorithm' activity sheet to record their predictions, the mats from Lesson 2, as well as the paper-bot cut-outs. Ask pupils to carefully move the paper-bot around the mat following the commands shown. Tell them to mark paper-bot's final (stop) position on their activity sheets, and explain that this is their prediction.</td></tr>
<tr><td>Success Criteria:
I can follow a sequence
I can predict the outcome of a sequence
I can compare my prediction to the program outcome</td><td>Suggested resources:
Flipchart

Beebots

Floor mats</td></tr>
</table>

		Once pupils have completed this, use the Bee-bots to test if their predictions were correct. **Independent:** In this activity, pupils will predict the outcome of four randomly created programs. Highlight that algorithms and programs are predictable, and that pupils will make predictions based on following a program; they will not be making guesses. Explain the steps: 1. First, pupils should select a 'start' square and direction for their robot. 2. Then, pupils should turn over four cards from a pile of mixed up command cards. 3. Pupils should record the commands in the box under the grid on their activity sheet. 4. Pupils should follow the program using paper-bot and record their prediction on their activity sheet. 5. Finally, pupils should test their prediction using a floor robot. Ask pupils to do this activity four times, and to fill in all four sections of their activity sheet. Tell pupils that if a prediction is wrong, they should re-enter the program into the robot to check that they haven't made a keying error. **Challenge** - If appropriate, allow pupils to increase the number of command cards that they turn over, and increase the length of the program that they need to make a prediction about. **Deepen:** Bring pupils back to the carpet to discuss their findings. Q – how were your predictions? Q – what did you find the most challenging? **Reflection / Assessment:** Metacognition – How do you feel about using logical reasoning to predict the outcome of a program?
Vocabulary: sequence, prediction, program, algorithm		

Lesson Number - 3

<table>
<tr><td>Key learning:
To design an algorithm.</td><td>Concepts:
Programs
Computational Thinking</td><td rowspan="2">Lesson structure: Introduction, direct teaching, activities, key questions

Engage:
Pupils to practise giving instructions to achieve a desired outcome. This could link to wider topic learning or a fun partner activity.

Introduce:
Recap Q – what is an algorithm?
When we give a set of instructions it is called an '<u>algorithm</u>'.
An algorithm is a precise set of instructions, which can be turned into a code by the robot or computer.

Tell pupils that they will be designing a mat for the floor robots. Tell pupils that they will need to make careful decisions about where they put things in their mat design.
Ask pupils to pick a theme for their mats. (E.g. playground, theme park, seaside) Pupils will need to think of 6 pictures that link to their theme. Alternatively, you could ask pupils to relate their mat design to a current class topic. Ask pupils to talk to a partner for a minute about ideas for the theme of their mat.

Note: Alternatively, you could discuss the theme as a class and create a class list of suitable pictures.

P&C:
Hand out the blank mats, and explain that pupils will need to choose six squares into which they will add their pictures. The six pictures should be spread out and spaces should be left between them. Tell pupils that later in the lesson, some obstacles will be added in the spaces.

Pupils can then place 3 obstacles on their mats. These could be made as stand up cards so that they are not fixed in place initially and pupils can test their ideas. The obstacles need to be placed carefully so that the robot can still move around.
Show pupils that the obstacles need to be placed in spaces, and not in squares where there is a drawing.

Scaffolding opportunity – choose a class theme and a selection of appropriate pictures and obstacles that match the theme for pupils to use.

Independent:</td></tr>
<tr><td>Success Criteria:
I can explain the choices I made for my mat design
I can identify different routes around my mat
I can test my mat to make sure that it is usable
I can explain what my algorithm should achieve
I can create an algorithm to meet my goal
I can use my algorithm to create a program</td><td>Suggested resources:
Flipchart

Beebots

Floor mats

Whiteboards and pens</td></tr>
</table>

		Pupils need to choose a start square and a finish square. Pupils can use a whiteboard to create an algorithm to get their Bee-bot from the start to the end without bumping into an obstacle. Pupils test this algorithm using the Bee-bots. When they have planned and tested their route, tell them to try another route. Once pupils are happy with their design, they can stick their obstacles down so they can be used during the next lesson. **Scaffolding opportunity –** Pupils work in mixed ability groups to support each other. **Deepen:** Ask pupils to reflect on the activities that they have completed in the lesson, and whether they found anything that didn't go as expected. Introduce the term 'debugging'. Explain that programmers don't always get things right the first time, and when a programmer finds a problem in their program or algorithm, they fix it. Q – did you have to fix any of the algorithms that you generated? Q – what did you do? This can just be class discussion to introduce pupils to the idea of debugging. **Reflection / Assessment:** Metacognition **–** how do you feel about designing and testing an algorithm? **Note –** You will need to keep their mat designs as they will be reusing them in the next lesson.
Vocabulary: algorithm, test, design, route, debug		

Lesson Number - 4		
Key learning: To create and debug a program that I have written.	**Concepts:** Programs Computational Thinking	**Lesson structure: Introduction, direct teaching, activities, key questions** **Engage:**

		Tell pupils to think back to the end of the last lesson, and ask if anyone can remember what 'debugging' is. Q – what is debugging? Remind pupils that algorithms and programs may not be right first time, and if something isn't right with an algorithm or program, the problem (or 'bug') needs to be found and fixed. This is called 'debugging'.
Success Criteria: ● I can plan algorithms for different parts of a task ● I can test and debug each part of the program ● I can put together the different parts of my program	**Suggested resources:** Flipchart Beebots Floor mat designs from previous lesson Whiteboards and pens	**Introduce:** Model an example of debugging. Show and discuss an algorithm on the flipchart as a class. Identify where the Bee-bot is supposed to finish. Use arrows to show a planned route. Then display the algorithm underneath. Use a paper bot to follow the algorithm to demonstrate where the Bee-bot would finish. It should reveal that the robot will not move to the correct square. Ask pupils to identify where the algorithm is wrong. **P&C:** Give pupils a series of algorithms on a sheet that they need to debug. Tell pupils that they need to follow each algorithm, and mark where the robot would get to with each algorithm. Explain that if the robot wouldn't get to the intended destination (the highlighted square), they should identify where the algorithm is wrong. Work in pairs. **Independent:** In the previous lesson, pupils wrote algorithms to get from one square to another. In this lesson, they need to visit two squares. Tell pupils that in their pairs, they need to agree a starting position and two squares to visit. Tell pupils that because the task is more complicated in this lesson, they will plan and test it a section at a time. Explain that pupils will be creating a larger algorithm (and program) in two stages. Ask pupils to first design an algorithm to move the robot from its starting square to the first destination. Ask pupils to draw their first algorithm on a whiteboard and check it by following it, as they have done in previous lessons. Once they have followed it, ask them to try it as a program for the floor robot. They should run the program and debug their algorithm if necessary. As in previous lessons, if the robot doesn't move as expected, they should clear the program and re-enter it, as a first debugging check. Once pupils have got their first algorithm right, tell them to move on to the second stage. Ask them to set aside the successful algorithm for the first stage to be reused later. Ask them to then place the robot on the first destination square, facing in the same direction as it did when the program for the first stage

		finished running, and design the algorithm to get to the final square. Again, pupils will need to test and debug where necessary. **Note:** It's important that pupils start the second stage of planning with the robot on the first destination square, facing in the same direction as it did when the program for the first stage finished running. Pupils should now have two algorithms that they have tested and debugged. Tell them to place their robot back at the starting position, enter their first algorithm, then press 'Pause', and then enter their second algorithm. Once they have entered their program, they should press 'Go' and run it. Their robot should visit the first square, pause, and then move on to the second square. Remind them that if the robot doesn't move as expected, they need to clear the robot's program memory and re-enter their program. **Deepen:** Pupils can take a video of their successful program and upload it to their Seesaw account. Q – Why do you think programmers break down complicated tasks down into chunks? Doing this makes complicated tasks easier, and makes it easier to fix bugs. They design the program for each section, test, and debug it. Once the program is right, they add another section. **Reflection / Assessment:** Metacognition – how do you feel about creating and debugging a program?
Vocabulary: debug, algorithm, program		

Lesson Number - 5		
Key learning: I can explain what bullying is, how people may bully others and how bullying can make someone feel.	**Concepts:** Online safety	**Lesson structure: Introduction, direct teaching, activities, key questions** **Engage** Knowledge Map Pre-assessment for Year 2 unit on Online Bullying.

Success Criteria:	**Suggested resources:**	**Introduce:**
• Identify some characteristics that are typical of bullying behaviour (online and offline). • Consider the motives behind bullying behaviour. • Show awareness of the range of emotions that people involved in a bullying situation might feel. • Identify who people can turn to for help and support. • Identify some sources of support in different contexts.	Project Evolve resources – Year 2: Online Bullying Flipchart Worksheets / paper	Q – What is bullying? Ask learners to work in pairs to discuss and define what they think bullying is. Take some suggestions and discuss their definitions. Give definition of bullying - Bullying is repeated, negative behaviour that is intended to make others feel upset, uncomfortable or unsafe. Show video from the anti-bullying alliance. https://anti-bullyingalliance.org.uk/tools-information/what-bullying Q – has your definition of bullying changed? Discuss the difference between offline and online bullying. Online bullying can happen anywhere. Q – where do you think online bullying could take place? (Online games, messaging e.g. texts, chat rooms) Look at a bullying scenario. Show pictures of different emotions to support children. Ask pupils how both the victim and the person doing the bullying would be feeling. Ask them how they would feel as a bystander. **P&C:** In table groups, pupils work through some scenarios around online bullying and consider how each person in the situation might be feeling. ***Scaffolding opportunity*** **–** *word bank of emotions to support children to identify different emotions.* CHECKPOINT Give 2 different scenarios. Pupils to identify whether it is or isn't bullying. Refer back to definition of bullying to help pupils answer. **Independent:** Ask learners to consider who they would ask for help at school, at home or online if they were worried about something. Take suggestions and discuss as a group. Encourage learners to think of sources of support other than known adults (e.g. contacting a helpline or online chat, using a report button in a game or app, etc.).

		Use this opportunity to highlight the help provided by your school/youth group/organisation and ensure that learners are familiar with the ways they can seek help. Repeat the discussion but focus on who can help with problems around bullying (online or offline). Ask learners if any of the people/sources of support are different compared to the previous discussion. Can they explain why/why not? Ask learners if they would do something different to get help if they were facing online bullying as opposed to offline bullying. Can they explain why? Explore why people don't always seek help/support for bullying. Take suggestions then reveal some common reasons. These could include: • They blame themselves for what is happening to them. • They are worried that seeking help will make the bullying worse. • They confuse bullying with teasing or a joke. • They don't think adults will understand/know what to do to make it better. Learners to create their own *supportive sunflower*; a way to record all the people and sources of help they can turn to if they have any concerns about bullying. Provide each learner with a sunflower worksheet, or you can ask them to draw their own sunflower. Learners could write their name (or draw a picture of themselves) in the centre of the sunflower, then add a name of a person/source of support to each sunflower petal. **Deepen:** Learners to read out their sunflowers to the rest of the group. **Reflection / Assessment:** Remind learners of the importance of a support network. Sometimes they may find that telling one person isn't enough - they may underestimate the severity of an issue, not know how to help or not recognise that there is a problem. So continuing to tell people until they get the help they need is key!
Vocabulary: bullying, online, offline, emotions, victim, support		

Lesson Number - 6

<table>
<tr><td>Key learning: I can explain why anyone who experiences bullying is not to blame.</td><td>Concepts:
Online Safety</td><td rowspan="2">Lesson structure: Introduction, direct teaching, activities, key questions

Engage:
Recap what bullying is.

Introduce:
Q – what does the word accident mean?
Q – can you think of an example of an accident?

Q – what does the word intentional mean?
Q – what sort of things do we that are intentional, planned and on purpose?
Q – is bullying accidental or intentional?

P&C:
This is a game that could be done outside / in the hall with hoops. Alternatively you could do it in the classroom with signs.

Have 2 hoops / large pieces of paper. One labelled as ‘Blame’ the other ‘Not to blame’.
Read out scenarios and ask learners to stand next to the hoop / paper they feel is correct if they considered themselves in the situation. Once they have selected, ask learners to explain their decision.

Here are some examples:
1. Someone snatches something out of your hand.
2. Your friends are shouting at each other.
3. Your friend hits you.
4. You spill a drink.
5. Someone takes something that belongs to you without asking.
6. Someone tells a joke that makes fun of you.
Teacher to include own ideas tailored to the class / include learners’ ideas.

Independent:
Repeat the activity but with specific examples of online behaviour.

1. Someone posts an embarrassing picture of you for others to see.
2. Your friend teases you for losing in an online game.
3. Someone leaves comments under your video, saying unkind things about how you look.
4. Your friend blocks you when you try to say ‘hi’ in the chat.</td></tr>
<tr><td>Success Criteria:
Identify examples of bullying behaviour.
Recognise the difference between accidental and intentional behaviours that may affect others.
Explain reasons why the blame lies with those who display bullying behaviours, not the target.</td><td>Suggested resources:
Project Evolve Year 2 – Online Bullying resouces

Hoops / large pieces of paper.

Flipchart</td></tr>
</table>

		5. Someone sends you a picture/video that makes you feel worried or upset. 6. You have an argument with someone online and they threaten to find you and hurt you. *Teacher to include own ideas tailored to the class / include learners' ideas.* **Deepen:** Reminder learners that, if someone bullies you, it is not your fault, you are not to blame. The person doing the bullying has chosen to do this. You are the target of the bullying as it is happening to you. Q - If you are bullied - what can you do? Recap ideas from the previous lesson (support sunflowers) Explain the 'Be' Code: Be Brave - Stand up for yourself, you have the right to be safe. Be Firm - Be strong and ask them politely to stop, remember your manners. Be in control - Don't retaliate, walk away and tell an adult. Pupils could design posters individually / pairs / groups to remind others about the 'Be' Code. **Reflection / Assessment:** Complete knowledge map "assess impact" on the lesson.
Vocabulary: intentional, accidental, bullying, behaviour, blame		