|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Computing** | **Computing Systems and Networks – Technology around us.**  **Skills**  Explain technology as something that helps us.  Locate examples of technology in the classroom and explain how these help us.  Name the main parts of a computer.  Switch on and log in to a computer.  Use a mouse to click, drag, open a programme, create a picture, click and drag to make objects on a screen.  Tell you that writing on a computer is called typing.  Type their name on a computer.  Save to and open from a file.  Use arrow keys to move the cursor.  Delete letters.  Identify rules to keep us safe and healthy when using technology in and beyond the home.  Give examples of some of these rules and how we benefit from them.  **Knowledge**  Children need to understand the definition of technology as something that has been made by people to help us.  Understand objects which and are not examples of technology.  Children need to have a knowledge of a variety of computer mouse and trackpad devices and explain the different functions they perform.  Children will experience use of a painting programme.  Children will have knowledge of the basic functions of a computer keyboard.  Children should be familiar with any school rules which may be in place.  **Vocabulary**  Technology, computer, mouse, trackpad, keyboard, screen, click, drag, draw, double-click, click and drag, Input device, shift, spacebar, capital letter, full stop, safely, responsibly. | **Creating Media – Digital Painting**  **Skills**  Make marks and draw lines on a screen and explain which tools where used.  Use paint tools to draw a picture.  Make marks with square and line tools.  Use shape and line tools effectively and use these to recreate the work of an artist.  Choose appropriate shape and colour choices.  Create a picture in the style of an artist.  Know that different paint tools do different jobs.  Choose appropriate paint tools and colours to recreate the work of an artist and say which tools where helpful and why.  Make dots of colour on a page and create a picture on the style of an artist on my own.  Change the colour and brush size.  Explain that pictures can be made in lots of different ways.  Spot the differences between painting on a computer or using paper and say which I prefer.  **Knowledge**  Children will need to be familiar with free hand-painting tools for digital painting programme.  Be familiar with the style of Piet Mondrian and Henri Matisse  Be familiar with primary colours; and the line, shape, fill and undo tools in a digital painting programme.  Children need to be familiar with the following painting tools in a digital programme: Paintbrush, pencil, fill, erase, undo, shape, brush styles.  **Vocabulary**  paint programme, tool, paintbrush, erase, fill undo, Primary colours, shape tools, line tool, fill tool, undo tool, feelings, colour, brush style, brush size, pointillism, pictures, painting, computers, like, prefer, dislike.  **Artists:** Henri Matisse, Wassily Kandinsky, Piet Mondrian, George Seurat | **Creating Media – Digital Writing**  **Skills**  Open a word processor  Recognise, identify and find keys on a keyboard.  Enter text into a computer  Use letter number and space keys and backspace to remove text.  Type capital letters  Explain what keys do  Identify the toolbar and use bold, italic and underline.  Select a word by double clicking and select all of the text by clicking and dragging.  Change fonts  Say what tool I used to change the text and decide if my changes have improved my writing.  Use undo to remove changes.  Write a message on a computer and on paper, compare these and say which I like best.  **Knowledge**  Children will need to be familiar with a word processor and specific sectors of the toolbar.  **Vocabulary**  Word processor, keyboard, keys, letters, Microsoft Word, Google Docs, numbers, space, backspace, text cursor, toolbar, capital letters, bold, italic, underline, select, font, mouse, undo | **Grouping Data**  **Skills**  Describe objects using labels and match objects to groups by identifying the label for a group.  Count and group objects  Describe and object and its property and find objects with similar properties.  Group similar objects and group objects in more than one way.  Count how many objects share a property.  Choose and describe groups of objects and record how many objects are in a group.  Describe how to group objects to answer a question.  Compare groups of objects and record and share what I have found.  **Knowledge**  A key concept throughout this unit is understanding that computers are not intelligent. Though, they may seem like they are able to complete tasks autonomously, they are using input from humans, for example searching for images that have been labelled by a human, or ‘counting’ data that has been grouped by humans.  The term ‘object’ is used to describe anything that can be labelled with properties, eg animals, pencils, or trees.  Objects are named to make it easier for humans to know what other humans are talking about, eg ‘tree’. The name may change depending on context (sometimes ‘tree’ is enough, but sometimes ‘oak tree’ may be required), but it is always a property that an object can be labelled with. A property describes an object. A label is a property used to describe an object, eg ‘green’. This is the data that is collected about the object.  Labelling, grouping, and searching are important aspects of data and information.  Searching is a common operation in many applications, and requires an understanding that to search data, it must have labels.  Data set is used to describe a collection of related data.  **Vocabulary**  Object, label, group, search, image, property, colour, size, shape, value, data set, more, less, most, fewest, least, the same. | **Programming A – Moving a robot**  **Skills**  Predict the outcome of a command on a device  Match a command to an outcome  Run a command on a device  Follow an instruction and give directions  Recall words that can be acted out.  Compare forwards and backwards movements  Start a sequence from the same place.  Predict the outcome of a sequence involving forwards and backwards commands.  Compare left and right turns.  Experiment with turn and movement commands to move a robot.  Predict the outcomes of a sequence involving up to four commands.  Explain what my programme should do.  Choose the order of commands in a sequence  Debug my programme  Identify several possible solutions  Plan two programmes and use these to get to the same place.  **Knowledge**  Ensure you are familiar with your school’s floor robots, including charging or battery requirements. You should also know how to switch the device on and off, as well as key functions such as clearing memory.  On Bee-Bot and Blue-Bot, the Go button starts a program running, but it will also stop a program while it’s running.  Children should be aware of how words can be used for instructions and how those words might be used in the activities. For example, the instruction ‘walk’ starts a process without an end. A human might ask how far they should go, or they may stop if they encounter an obstacle. If a robot could be issued with the command ‘walk’, it would start a continuous process that wouldn’t be stopped and could cause the robot to walk into obstacles. A more suitable instruction would be ‘step’.  Be familiar with forwards, backwards, clear and go buttons on the floor robot.  Start from the same square on the mat to make the outcome of the robot programme repeatable and predictable.  Be familiar with the term commands and how they are implemented on the device.  Understand that most floor robots turn left or right on the spot.  Children write and test, they should always start the robot from the same square, pointing the same way  Press clear before entering each program  An algorithm is a precise set of ordered instructions, which can be turned into code. An algorithm is a part of the design of the program; it is not the program itself. After an algorithm has been designed, it can be implemented on a computer as code. In floor robot terms, the algorithm is explaining precisely where you want the robot to go; this is then turned into ‘code’ by pressing the buttons.  Children plan their route before creating their programme.  **Vocabulary**  Forwards, backwards, turn, clear, go, commands, instructions, directions, left, right, plan, algorithm, programme, route | **Programming B – Programming Animations**  **Skills**  Find and use the commands to move a sprite  Compare different programming tools  Use more than one block by joining them together  Use a start block in a programme  Run a programme  Find blocks that have number and change the value and say what happens when I change the value.  Show that programme can include more than one sprite  Delete a sprite and add blocks to each sprite.  Choose appropriate artwork for the project  Decide how each sprite will move and create an algorithm for each sprite.  Use sprites that match my design.  Add programming blocks based on my algorithm  Test the programmes I have created.  **Knowledge**  Understanding ScratchJr, including the ability to join blocks together and run programs using Start blocks. Change backgrounds and delete sprites. Each of these skills is supported in the slides.  An algorithm is a precise set of ordered instructions, which can be turned into code. An algorithm is a part of the design of the program; it is not the program itself. After an algorithm has been designed, it can be implemented on a computer as ‘code’. In Scratch terms, the algorithm explains precisely where you want the sprite to go; this is then turned into code by using the programming blocks.  Know which block categories hold blocks with values underneath and how to change these values.  Add and delete sprites and add programmes to each sprite.  Explore the design and code levels.  Understanding of Speed blocks and how to edit sprites.  Explore moving between the ‘task’ and ‘design’ levels of the project.  ‘Design’ in programming means choosing any sprites that are needed, creating/choosing artwork for the sprites and backgrounds, and writing algorithms so that learners are ready to move to the ‘code’ level where they will implement all of their choices.  Know how to re-size sprites  **Vocabulary**  Scratch junior, beebot, command, sprite, compare, programming, programming area, block, joining, command, start block, run, background, delete, reset, algorithm, predict, effect, change, value, instructions, sprite, delete, appropriate. |