|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Computing** | **Connecting Computers**  **Skills**  Explain that digital devices accept inputs.  Explain that digital devices produce outputs.  Follow a process.  Classify input and output devices.  Model a simple process.  Design a digital device.  Explain how to use digital devices for different activities.  Recognise similarities between using digital devices and non-digital tools.  Suggest differences between using digital devices and non-digital tools.  Recognise different connections.  Explain how messages are passed through multiple connections.  Discuss why we need a network switch.  Recognise that a computer network is made up of multiple devices.  Demonstrate how information can be passed between devices.  Explain the role of a switch, server, and wireless access point in a network.  Identify how devices in a network are connected to one another.  Identify the benefits of computer networks.  **Knowledge**  Understand digital and non-digital devices.  Be familiar with the inputs and outputs of a range of digital devices.  Use the functions on a digital paint programme.  Understand how information flows around a computer network.  Have knowledge of a simple school network.  Be familiar with how the main devices on a school network are connected to one another and know where they are located.  **Vocabulary**  Digital device, input, output, process, programme, connection, network, network switch, server, wireless access point (WAP) | **Stop-frame animation**  **Skills**  Draw a sequence of pictures.  Create an effective flip-book style animation.  Explain how an animation/flip-book works.  Predict what an animation will look like.  Explain why little changes are needed for each frame.  Create an effective stop-frame animation.  Break down a story into settings, characters and events.  Describe an animation that is achievable on screen.  Create a storyboard.  Use onion skinning to make small changes between frames.  Review a sequence of frames to check work.  Evaluate the quality of their animation.  Explain ways to make their animation better.  Evaluate another learner’s animation.  Improve their animation based on feedback.  Add another media to their animation.  Explain why another media was added to their animation.  Evaluate their final film.  **Knowledge**  Understand how to make a simple flip-book animation.  Understand how to make a simple stop-frame animation.  Understand what the iMotion app is capable of.  Know how to make a simple stop frame animation in iMotion and how to keep consistency in animations.  Understand ways animations can be improved.  Understand how in import a video from a camera roll into iMovie and add music, titles etc.  **Vocabulary**  Animation, flip book, stop-frame animation, frame, sequence, image, photograph, setting, character, events, onion skinning, consistency, delete, media, import, transition | **Desktop Publishing**  **Skills**  Explain the difference between text and images.  Recognise that text and images can communicate messages.  Identify the advantages and disadvantages of using text and images.  Change font size, style and colour for a given purpose.  Edit text.  Explain that text can be changed to communicate more clearly.  Define the term ‘page orientation’.  Recognise placeholders and say why they are important.  Create a template for a purpose.  Choose the best locations for context.  Paste text and images to create a magazine cover.  Make changes to content after it has been added.  Identify different layouts.  Match a layout to a purpose.  Identify the uses of desktop publishing in the real world.  Say why desktop publishing might be helpful.  Compare work made on desktop publishing to work created by hand.  **Knowledge**  Understand the advantages of using, text, images or both.  Know how to change the font, style and colour.  Understand the terms template, placeholder and orientation.  Know how to use the tools in Adobe Spark.  Understand the different way in which information can be laid out on a page.  Understand places desktop publishing is used in the real world.  **Vocabulary**  Text, images, advantages, disadvantages, communicate, font, font style, template, landscape, portrait, orientation, placeholder, layout, content, desktop publishing, copy, paste, purpose , benefits | **Branching Databases**  **Skills**  Investigate questions with yes/no answers.  Make up a yes/no question about a collection of objects.  Create two groups of objects separated by one attribute.  Select an attribute to separate objects into groups  Create a group of objects within an existing group.  Arrange objects into a tree structure.  Select objects to arrange in a branching database.  Group objects using my own yes/no questions.  Prove my branching database works.  Crete yes/no questions using given attributes.  Explain that questions need to be ordered carefully to split objects into similarly sized groups.  Compare two branching database structures.  Select a them and choose variety of objects.  Create questions and apply them to a tree structure.  Use my branching database to answer questions.  Explain what a pictogram and branching database tells me  Compare two ways of presenting information.  **Knowledge**  Know how to describe objects using appropriate attributes.  Know what a branching database is.  Know how to use a branch tool.  Know an attribute describes objects.  Know the purpose of pictograms and branching databases  **Vocabulary**  Attribute, value, questions, table, objects, branching database, database, equal, even, separate, structure, compare, order, organise, value, J2Data, selecting, information, decision tree, | **Programming A – Sequencing Sounds**  **Skills**  Identify the objects in a scratch project.  Explain that objects in scratch have attributes  Recognise that commands in Scratch are represented as blocks.  Identify that each sprite is controlled by the commands that are chosen.  Choose a word which describes an on-screen action for the design.  Create a program following a design.  Start a program in different ways.  Create a sequence pf connected commands.  Explain that the objects in the project will respond exactly to the code.  Explain what a sequence is.  Combine sound commands.  Order notes into a sequence.  Build a sequence of commands.  Decide the actions for each sprite in a program.  Make design choices for my artwork.  Identify and name the objects needed for the project.  Relate a task description to a design.  Implement my algorithm as code.  **Knowledge**  Be familiar with the Scratch Programming environment  Move between the design and code levels of the project.  Code multiple sprites in one project and be aware of new blocks.  Be familiar with three event blocks which cause an event to start.  Move between four levels of abstraction.  Know the difference between different blocks.  Be aware of sounds attached to certain sprites in scratch.  Be familiar with the concept of costumes for sprites and backdrops for the stage.  Work on all four levels of abstraction. – task – what is needed? Design – what it should do?  Code – how it is done?  Running the code – what it does.  Understand that an algorithm is a precise set of ordered instructions.  Use an octave scale to mimic the behaviour of a piano.  **Vocabulary**  Scratch, programming, blocks, commands, code, sprite, costume, stage, backdrop | **Programming B – Events and actions in programs**  **Skills**  Explain the relationship between an event and an action.  Choose which keys to use for actions and explain the choices.  Explain a way to improve a program.  Choose a character for my project.  Choose a suitable size for a character in a maze.  Program movement.  Chose a programming extension.  Consider the real world when making design choices.  Choose blocks to set up the program.  Identify additional features.  Choose suitable keys to turn on additional features.  Build more sequences of commands to make the design work.  Test a program against a given design.  Match a piece of code to an outcome.  Modify a program using design.  Make design choices and justify them  Implement the design.  Evaluate the project.  **Knowledge**  Awareness of basic movements in scratch. Know how to program multiple sprites.  Apply skills in a different context.  Move from task and design levels to design code and running levels.  Be familiar with extension blocks in scratch.  Set up projects to start in the same way every time they run.  Save projects and the end of each lesson.  Be familiar with pen extension blocks.  Change the colour of the pen.  Know the difference between blocks.  Be familiar with the term debug and the process of debugging.  Set pen size block.  Apply skills and concepts.  **Vocabulary**  Motion, sprite, event, logic, move, resize, algorithm, extension block, pen up, set up, pen, design, action, debugging, errors, setup, code, test, debug |