Swain House Primary School
Computing Progression of Skills

Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
Autumn 1 - Computing System and Networks					
Technology Around Us - Choose a piece of technology to do a job. - Recognise that some technology can be used in different ways. - Identify the main parts of a computer. - Use a mouse in different ways. - Use keyboard to type. - Use a keyboard to edit text. - Use the keyboard to edit text. - Show how to use technology safely.	**IT Around Us** - Describe some uses of computers. - Identify information technology in school. - Identify information technology beyond school. - Show how to use information technology safely.	**Connecting Computers** - Identify input and output devices. - Explain that a computer system accepts an input and processes it to produce an output. - Explain how a computer network can be used to share information. - Explain the role of a switch, server, and wireless access point in a network. - Identify network devices around me. - Explain how networks can be connected to other network.	**The Internet** - Describe how networks connect to other networks. - Recognise the need for security on the internet. - Describe the types of content/media that can be added, created, and shared on the World Wide Web. - Access the World Wide Web. - Describe the benefits and limitations of the World Wide Web.	**Sharing Information** - Describe the input and output of a search engine. - Demonstrate that different search terms produce different results. - Evaluate the results of search terms.	**Internet Communication** - Recall how to use a search engine. - Compare the results from different search engines. - Demonstrate that different search terms produce different results. - Explain that search terms need to be chosen carefully. - Evaluate the results of search terms. - Identify that results from search engines can include adverts. - Identify different ways to communicate without technology. - Evaluate different methods of online communication.
Autumn 2 – Creating Media					
Digital Painting - Create a picture using freehand tools. - Use shape and line tools when precision is needed. - Use a range of paint colours. - Use the fill tool to colour an enclosed area. - Use the undo button to correct a mistake. - Combine a range of tools to create a piece of artwork.	**Digital Photography** - Capture a digital image. - Take photographs in both landscape and portrait format. - View photographs on a digital device. - Decide which photographs to keep. - Use filters to edit the appearance of a photo. - Hold the camera still to take a clear photo. - Use zoom to change the composition of a photo. - Consider lighting before taking a photograph. - Improve a photograph by retaking it.	**Stop- Frame Animation** - Plan an animation using a storyboard. - Set up a work area with an awareness of what will be captured. - Capture an image. - Use the onion skinning tool to review subject position. - Move a subject between captures. - Review a captured sequence of frames as an animation. - Remove frames to improve an animation. - Add media to enhance an animation. - Review a completed project.	**Audio Production** - Record sound using a computer. - Play recorded audio. - Import audio into a project. - Delete a section of audio. - Change the volume of tracks in a project.	**Vector Drawing** - Add an object to a vector drawing. - Select one object or multiple objects. - Delete objects. - Move objects between the layers of a drawing. - Duplicate objects using copy and paste. - To modify objects. - Reposition objects. - Group and ungroup selected objects. - Combine options to achieve a desired effect. - Create a vector drawing for a given purpose.	**3D Modelling** - Position 3D shapes relative to one another. - Use digital tools to modify 3D objects. - Combine objects to create a 3D digital artefact. - Use digital tools to accurately size 3D objects. - Construct a 3D model which reflects a real world object.

Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
Spring 1 – Programming A					
Moving a Robot	**Robot Algorithms**	**Sequencing Sounds**	**Repetition in Shapes**	**Selection in Physical Computing**	**Variables in Games**
• Enact a given word. • Predict the outcome of a command on a device. • List which commands can be used on a given device. • Choose a command for a given purpose. • Choose a series of commands that can be run as a program. • Build a sequence of commands in steps. • Combine commands in a program. • Run a program on a device. • Run a command on a floor robot.	• Choose a series of words that can be enacted as a sequence. • Choose a series of instructions that can be run as a program. • Create a program • Trace a sequence and make a prediction. • Run a program on a device. • Debug a program that I have written.	• Build a sequence of commands. • Combine commands in a program. • Order commands in a program. • Create a sequence of commands to produce a given outcome.	• List an everyday task as a set of instructions. • Use an indefinite loop to produce a given outcome. • Use a count-controlled loop to produce a given outcome. • To plan a program that includes appropriate loops to produce a given outcome. • Recognise tools that enable more than one process to be run at the same time (concurrency). • Create two or more sequences that run at the same time.	• Create a condition-controlled loop. • Use a condition in an 'if…then…' statement to start an action. • Use selection to switch the program flow in one of two ways. • Use a condition in an 'if…then…else…' statement to produce given outcomes.	• Identify a variable in an existing program. • Experiment with the value of an existing variable. • Choose a name that identifies the role of a variable to make it easier for humans to understand it. • Decide where in a program to set a variable. • Update a variable with a user input. • Use an event in a program to update a variable. • Use a variable in a conditional statement to control the flow of a program. • Use the same variable in more than one location in a program.
Spring 2 - Data and Information					
Grouping Data	**Pictograms**	**Branching Databases**	**Data Logging**	**Flat-file Databases**	**Introduction to Spreadsheets**
• Identify some attributes of an object. • Collect simple data. • Show that collected data can be counted. • Describe the properties of an object. • Choose an attribute to group objects by. • Group objects to answer questions. • Explain that objects can be grouped by similarities. • Describe a group of objects.	• Recognise that people, animals and objects can be described by attributes. • Show I can enter data onto a computer. • Use a computer to view data in different formats. • Use pictograms to answer simple-attribute questions. • Use a computer to answer comparison questions (graphs, tables).	• Create questions with yes/no answers. • Choose questions that will divide objects into evenly sized subgroups. • To repeatedly create subgroups of objects. • To identify an object using a branching database. • To retrieve information from different levels of branching database.	• Use a digital device to collect data automatically. • Choose how often to automatically collect data samples. • Use a set of logged data to find information, • Use a computer program to sort data by one attribute. • Export information in different formats.	• Choose different ways to view data. • Choose which attribute and value to search by to answer a given question (operands) • Ask questions that need more than one attribute to answer. • Choose which attribute to sort data by to answer a given question. • Choose multiple criteria to search data to answer a given question (AND and OR). • Select an appropriate graph to visually compare data. • Choose suitable ways to present information to other people.	• Calculate data using a formula for each operation. • Use functions to create new data. • Use existing cells within a formula. • Choose suitable ways to present spreadsheet data.

Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
Summer 1 – Creating Media					
Digital Writing • Use letter, number, and space keys to enter text into a computer. • Use punctuation and special characters. • Use backspace key to remove text. • Position the text cursor in a chosen location. • Select text. • Choose options to achieve a desired effect. • Change the appearance of text on a computer. • Use undo.	**Making Music** • Experiment with musical patterns on a computer. • Experiment with different sounds on a computer. • Use a computer to create a musical pattern. • Use a computer to compose a rhythm. • Use a computer to play the same music in different ways (e.g. tempo). • Evaluate a musical composition created on a computer. • Improve a musical composition created on a computer.	**Desktop Publishing** • Show that page orientation can be changed. • Add text to a placeholder. • Edit text in a placeholder. • Organize text and image placeholders in a page layout. • Add and remove images to and from placeholders. • Move, resize and rotate objects. • Choose fonts and apply effects to text. • To review a document.	**Photo Editing** • Change the composition of a digital image by rotating and flipping. • Change the composition of a digital image by cropping. • Adjust colours of a digital image. • Apply filters to a digital image. Apply effects to a digital image. • Use clone, copy, and paste to change the composition of a digital image. • Use cloning to retouch a digital image. • Add text to a digital image.	**Video Editing** • Use different camera angles. • Use pan, tilt and zoom. • Identify features of a video recording device or application. • Combine filming techniques for a given purpose. • To determine what scenes will convey your idea. • Decide what changes I will make when editing. • Choose to reshoot a scene or improve later through editing. • Use split, trim and crop to edit a video.	**Web Page Creation** • Review an existing website (navigation bars, header). • Create a new blank web page. • Add text to a web page. • Set the style of text on a web page. • Change the appearance of text. • To embed media in a web page. • Add web pages to a website. • Preview a web page (different screen sixes) • Insert hyperlinks between pages. • Insert hyperlinks between pages. • Insert hyperlinks to another site.
Summer 2 – Programming B					
Introduction to Animation • Choose a series of words that can be enacted as a program. • Choose a series of commands that can be run as a program. • Run a program on a device.	**An Introduction to Quizzes** • Choose a series of words that be enacted as a sequence. • Explain what happens when we change the order of instructions. • Choose a series of commands that can be run as a program. • Trace a sequence to make a prediction. • To test a prediction by running the sequence. • To create and debug a program I have written. • To run a program on a device.	**Events and Actions in Programs** • Build a sequence of commands. • Combine commands in a program. • Order commands in a program. • Create a sequence of commands to produce a given outcome.	**Repetition in Games** • List an everyday task as a set of instructions including repetition. • Use an indefinite loop to produce a given outcome. • Use a count-controlled loop to produce a given outcome. • Plan a program that includes appropriate loops to produce a given outcome. • Recognise tools that enable more than one process to be run at the same time (concurrency). • Create two or more sequences that run at the same time.	**Selection in Quizzes** • Choose a condition to use in a program. • Create a condition-controlled loop. • Use a condition in an 'if...then...' statement to start an action. • Use selection to switch the program flow in one of two ways. • Use a condition in an 'if...then...else...' statement to produce given outcomes.	**Sensing** • Identify a variable in an existing program. • Experiment with the value of an existing variable. • Choose a name that identifies the role of a variable to make it easier for humans to understand it. • Decide where in a program to set a variable. • Update a variable with a user input. • Use an event in a program to update a variable. • Use a variable in a conditional statement to control the flow of a program. • Use the same variable in more than one location in a program.