Stage	Skills	Resources	Summary
Lower School			
P1	● Classifying objects ● Identifying computing devices (hidden examples e.g. automatic doors) ● Following directional instructions ● Designing a simple sequence of directional instructions ● Identifying and correcting errors in sets of instructions (de-bug).	**Grid mats** **Direction cards** **Beebots** **Activ Inspire** **Kodable**	The children will be introduced to the concept of programming through several unplugged activities. This will begin with identifying steps and patterns in everyday processes. Pupils should be able to identify what will happen if steps within a process change or are missing. Pupils will then be introduced to directional instructions through a variety of unplugged, practical activities, games on Activ Inspire, Beebots and a bee-bot website. https://www.bee-bot.us/emu/beebot.html The primary focus will be to ensure children have a firm understanding of directional signs and instructions (forward, back, left and right). Finally, children will be able to progress onto the first stage of the Kodable website "Smeeborg - Sequence Sector". This will allow them to apply their knowledge of programming by creating a sequence of directional instructions.
P2	● Constructing simple sequences ● Conditioning ● Loops ● Functions	**Kodable**	The children will be introduced to coding on a computer program. Children will be given a class code to access Kodable resources. They will begin by sequencing steps in a process. The concept of conditioning will then be introduced in which the children have to alter a simple sequence. The children will then be briefly introduced to loops and functions. Children will then create their own maze in the "create" section of Kodable.
P3	● Create simple sequences ● Make decisions based on logical thinking including If, AND, OR and NOT ● Demonstrates an understanding of the meaning of individual instructions when using a visual programming language. ● Constructs a sequence of instructions to solve a task explaining the expected output from each step. ● Identified when a program does not do what's intended and can correct errors/ bugs.	**Code.org** Course A (4-7 years/B (5-8) https://code.org/student/elementary https://codeclubprojects.org/en-GB/scratch/	Children will be introduced to coding through a series of lessons on '**code.org'**. Some lessons are 'unplugged' and therefore teach the concept without being on the computer. Each lesson focuses on a particular skill and increases in difficulty as you work through the lessons More able pupils may move onto to use 'Scratch' learning basic skills using the Scratch Card lessons (see link). Then they can complete Projects 1 - 3 on the 'Code Club' website (link shown in resources column).
Middle School			

P4	● Writing simple programs ● Animations ● Making a game	**Scratch**	Children will be introduced to Scratch. They will begin with writing a simple program and progress onto adding background and sounds, telling a story, making animating and moving their own sprite and making a game. Following this children will reinforce their learning through Scratch Projects.
P5	● Making simple quizzes ● Making games	**Scratch** https://codeclubprojects.org/en-GB/scratch/	Progression of Scratch programming learning. It increases in complexity chronologically and mostly comprises of game based creation scratch programming activities. 1. Making a simple quiz 2. Making a game where they catch dots 3. A save the earth game 4. An adventure game 5. A platform game 6. A memory game 7. A movement game 8. An image creator 9. A more complex quiz 10. A where's wally style game
Upper School		
P6	Creates programs in a visual programming language including variables and conditional repetition Identifies patterns in problem solving and reuses aspects of previous solutions appropriately, for example, reuse code for a timer, score counter or controlling arrow keys Identifies any mismatches between the task description and the programmed solution, and indicates how to fix them Explains the meaning of individual instructions (including variables and conditional repetition) in a visual programming language Predicts what a complete program in a visual programming language will do when it runs, including how the properties of objects for example, position, direction and appearance, change as the program runs through each instruction Explains and predicts how parallel activities interact Demonstrates an understanding that all computer data is represented in binary, for example, numbers, text, black and	**HTML & CSS** **Trinket** https://codeclubprojects.org/en-GB/resources/python-intro/	Intro in HTML (**HyperText Markup Language)**. The language used to make web pages. CSS (Cascade Styling Sheets) is responsible for the design or style of the website, including the layout, visual effects and background color. Writing HTML Writing CSS Children will learn about HTML lists and CSS colours. Children are introduced to CSS classes. They use multiple CSS classes to style text and also learn how to use background images and free Google fonts in their projects. Children are introduced to hyperlinks and embedding trinkets and Scratch projects within a webpage. Children will be introduced to linear and radial gradients in CSS. They will also learn more about borders and positioning. Children will to learn how to animate a simple scene using CSS. They will use the CSS @keyframes rule to animate various properties of images and divs. Children are introduced to multiple linked web pages in the same project each with their own CSS file. Children will learn how to create a two-column layout. They will also recap lots of the HTML & CSS that they have learned in other projects.

	white graphics. Describes the purpose of the processor, memory and storage and the relationship between them Demonstrates an understanding of how networks are connected and used to communicate and share information, for example, the internet Can design a webpage focusing on both style and functionality. Embedding trinkets and Scratch projects within a webpage. Learn how to animate a simple scene using CSS		Children will be introduced to JavaScript to change the colour of the pixels. They will also learn how to use HTML tables to create a grid of pixels. **HTML and CSS lessons** ● Recipe https://codeclubprojects.org/en-GB/webdev/ ● Mystery Letter https://codeclubprojects.org/en-GB/webdev/ ● Project Showcase https://codeclubprojects.org/en-GB/webdev/ ● Build a Robot https://codeclubprojects.org/en-GB/webdev/ ● Stickers https://codeclubprojects.org/en-GB/webdev/ ● Sunrise https://codeclubprojects.org/en-GB/webdev/ ● Linked Rooms https://codeclubprojects.org/en-GB/webdev/ ● Magazine https://codeclubprojects.org/en-GB/webdev/ ● Pixel Art https://codeclubprojects.org/en-GB/webdev/ **Python Lessons** ● Activity 1: Encrypting Files ● Activity 2: Ring of Firewalls ● Activity 3: Photo Detective ● Activity 4: Database Cleanup
P7	Creates programs in a visual programming language including variables and conditional repetition Identifies patterns in problem solving and reuses aspects of previous solutions appropriately, for example, reuse code for a timer, score counter or controlling arrow keys Identifies any mismatches between the task description and the programmed solution, and indicates how to fix them Explains the meaning of individual instructions (including variables and conditional repetition) in a visual programming language Predicts what a complete program in a visual programming language will do when it runs, including how the properties of objects for example, position, direction and appearance, change as the program runs through each instruction Explains and predicts how parallel activities interact Demonstrates an understanding that all computer data is	**Cyber Skills Lessons use of python (easy to follow step by step guides and video intro to most lessons)** **Python Projects** https://codeclubprojects.org/en-GB/python/	Digital photographs contain hidden information.You'll use Python code to extract evidence hidden inside photographs. Step into the shoes of a cyber criminal and use computer code to attempt to "crack" a massive collection of passwords encrypting and decrypting files using code modifying settings for firewall to protect yourself Metadata - info stored on cameras/gps/etc how to clean up a hacked database using the terminal Python They will create ASCII art (images from text) and perform calculations based on user input. using variables and conditional statements. loops This project teaches lists and using files. This project introduces dictionaries by creating a dictionary that maps from human-friendly colour names to hex codes

represented in binary, for example, numbers, text, black and white graphics. Describes the purpose of the processor, memory and storage and the relationship between them Demonstrates an understanding of how networks are connected and used to communicate and share information, for example, the internet Create an encryption program Extract information from photographs use code to protect basic firewall understanding and manipulating variables, lists and dictionaries.		children will learn how to make an encryption program, to send and receive secret messages with a friend. This project introduces iteration (looping) over a text string. functions Children collect data and use Pygal to display it using Pie Charts and Bar Graphs. manipulating dictionaries and lists. Children will work with JSON and Python data structures. This will be achieved by understanding and manipulating variables, lists and dictionaries. **Cyber Skills Lessons** ● Lesson: Cracking One Million Passwords ● Lesson: How To Rob A Bank ● Lesson: Every Picture Tells A Story ● Lesson: Defend The Hospital ● Lesson: Defend The Fire Service **Python Lessons** ● About Me https://codeclubprojects.org/en-GB/python/ ● Rock, paper, Scissors https://codeclubprojects.org/en-GB/python/ ● Turtle Race! https://codeclubprojects.org/en-GB/python/ ● Team Chooser https://codeclubprojects.org/en-GB/python/ ● Colourful Creations https://codeclubprojects.org/en-GB/python/ ● Secret Messages https://codeclubprojects.org/en-GB/python/ ● Modern Art https://codeclubprojects.org/en-GB/python/ ● Popular Pets https://codeclubprojects.org/en-GB/python/ ● RPG https://codeclubprojects.org/en-GB/python/ ● Where is the Space Station? https://codeclubprojects.org/en-GB/python/ ● Robo-Trumps https://codeclubprojects.org/en-GB/python/ ● CodeCraft https://codeclubprojects.org/en-GB/python/