Broadway Junior School

Subject: Computing Topic: Coding Year: 4 Term: Spring 1

NC Purpose of Study
A high-quality computing education equips pupils to use computational thinking and creativity to understand and change the world. Computing has deep links with mathematics, science and design and technology, and provides insights into both natural and artificial systems. The core of computing is computer science, in which pupils are taught the principles of information and computation, how digital systems work and how to put this knowledge to use through programming. Building on this knowledge and understanding, pupils are equipped to use information technology to create programs, systems and a range of content. Computing also ensures that pupils become digitally literate – able to use, and express themselves and develop their ideas through, information and communication technology – at a level suitable for the future workplace and as active participants in a digital world.

Intent	Implementation	Impact
Our vision at Broadway Junior School is that all pupils should enjoy Computing and become confident, passionate and responsible users of technology both inside and outside of school. Through interesting, relevant and inclusive lessons, they will be able to leave Year 6 with a good understanding of how computers and the Internet work, along with being able to complete tasks on them creatively, independently and reflectively for a range of purposes and audiences. The Computing curriculum at Broadway is designed to enable pupils to embrace and utilise new technologies through the application of essential knowledge, principles and concepts. Technology is everywhere and will play a pivotal part in students' lives, therefore, we want to model and educate our pupils on how to use technology positively, responsibly and safely. We want our pupils to be creators, not just consumers of content and our broad curriculum encompassing 'Coding & Computational Thinking', 'Information Technology', 'Computing Systems & Networks' and 'Online Safety' reflects this. We recognise that the best prevention for a lot of issues we currently see with technology and social media is through education.	Computing skills are taught explicitly at Broadway with every pupil receiving dedicated Computing lessons throughout each half-term. Each half-term has a focus on either Information Technology, Coding & Computational Thinking or Computing Systems & Networks with Online Safety lessons also included throughout each half-term. The skills acquired are then embedded throughout the curriculum. Learning is sequenced to build knowledge, skills and vocabulary. Throughout units of work teachers encourage children to make links between past learning and new content. We recognise prior learning and build on it with memorable learning experiences and provide targeted support where necessary. The units address the requirements of the National Curriculum and are broken down to cover all elements of the different strands of Computing. The progression of knowledge and skills statements build year on year to deepen and challenge our learners. We use Purple Mash as the core of our curriculum tied with Teach Computing and activities from Project Evolve to create a comprehensive curriculum providing full curriculum coverage and a variety of skills for our children.	The impact of our Computing curriculum is that we provide pupils with a set of skills to embed a lifelong love of learning and that they build on the knowledge and skills from previous learning. We ensure that every child can become a confident user of technology, while being able to use it to accomplish a wide variety of goals, both at home and in school. Children will have a secure and comprehensive knowledge of how technology works in the world around them and will develop their understanding of how to deal with online situations safely. Children will become confident global citizens. The knowledge organisers outline what we want the children to know within each unit. By nature Computing is going to be practical and hands on. We give the children the opportunity to record individually their responses to key questions. Teachers ensure that children understand and apply the correct vocabulary linked to different elements of the curriculum and this is also recorded within the floor books. Prior and future learning links are also included, so that there is a clear sequence in relation to the progression of skills and activities.

Key Vocabulary
Action, Alert, Algorithm, Background, Button, Code Blocks, Command, Co-ordinates, Debug/Debugging, Design, Event, Execute, Flowchart, If Statement, If/Else Statement, Input, Nest, Object, Prompt, Implement, Predict, Repeat, Run, Properties, Selection, Sequence, Timer, Variable, Value.

Knowledge	Skills
• To begin to understand selection in computer programming. • To understand how an IF statement works. • To understand how to use co-ordinates in computer programming. • To understand the 'repeat until' command. • To understand how an IF/ELSE statement works. • To understand what a variable is in programming. • To use a number variable. • To create a playable game.	• Children can explore different object types in 2Code. • Children can use a background and objects to create a scene. • Children can plan an algorithm for their scene and use 2Code to program it. • Children can create a program that includes an IF statement. • Children can interpret a flowchart that depicts an IF statement. • Children can make use of the X and Y attributes (properties) of objects in their coding. • Children can create a program that includes an IF statement. • Children can read code that includes repeat until and IF/ ELSE and explain how it works. • Children can create a program that includes an IF/ ELSE statement. • Children can interpret a flowchart that depicts an IF/ ELSE statement.

Broadway Junior School

Subject: Computing Topic: Coding Year: 4 Term: Spring 1

	• Children can explain what a variable is in programming. • Children can create and use variables when programming. • Children can read code that includes repeat until and IF/ ELSE and explain how it works. • Children can create a program that includes and IF/ ELSE statement. • Children can interpret a flowchart that depicts an IF/ ELSE statement.

	Learning Objectives/End Points				
To understand that computer programs work by following algorithms and to use simple planning tools. Lesson 1	To build structured programs using events and outputs and debug using logical reasoning. Lesson 2	To use selection and co-ordinates to control program behaviour. Lesson 3	To use repeat and timer commands to control how programs run. Lesson 4	To understand what a variable is in programming. Lesson 5	To create a game that keeps score. Lesson 6

Prior and Future Learning Links:
• Unit 3:1 (Coding): Flowcharts, Timers, Repeat, Code-test-debug process. Unit 5.1 (Coding): Efficient coding, simulating a physical system, Decomposition & abstraction, Friction & functions, Introducing strings, Text strings and concatenation. Unit 5.9 (External Devices): Program an external device, Program an external controller using inputs and outputs, Monitor real world conditions, Code-test-debug.

Careers	
Linking Curriculum Learning to Careers Discuss how the skills developed in this unit are used in a range of careers, including: Software Developer – Creates apps, websites, and computer programs. Game Developer – Designs and programs video games. Web Developer – Builds and maintains websites. Robotics Engineer – Programs robots to perform tasks. AI Engineer – Creates systems that use artificial intelligence. App Developer – Makes apps for phones and tablets. Cybersecurity Specialist – Uses coding and technology to protect computers and networks. Data Scientist – Uses programming to analyse data and find patterns. Computer Engineer – Works with computer hardware and software. Animation/Graphics Programmer – Uses code to create visual effects, animations, and computer graphics.	**Learning from career and labour market information** Introduce careers such as software developers, game designers, app developers, robotics engineers and software testers. Explore how these professionals use code to create, control and improve digital systems. **Addressing the needs of each pupil** Provide inclusive coding activities with visual prompts, flowcharts, modelling and step-by-step support. Offer additional challenge through more complex algorithms, variables and game features so all pupils can develop their skills. **Linking curriculum learning to careers** Link algorithms, selection, IF/ELSE statements, coordinates, repeat commands, timers and variables to the work of programmers and game developers. Highlight how testing and debugging help professionals create reliable programs and interactive games. **Encounters with employers and employees** Where possible, invite a parent, school staff member or visitor who works with technology to explain how they use programming, logical reasoning or digital tools to solve problems in their job.

Broadway Junior School

Subject: Computing Topic: Coding Year: 4 Term: Spring 1

Transferable skills Logical thinking: Planning algorithms and using sequence and selection to control program behaviour. Problem-solving: Using IF/ELSE statements, coordinates and variables to achieve specific outcomes. Creativity: Designing and creating playable games with interactive features. Resilience: Testing, debugging and improving programs when errors occur. Planning and organisation: Using flowcharts and algorithms to structure a program before coding. Attention to detail: Checking commands, conditions, values and variables. Evaluation: Predicting what code will do and reviewing whether a program works as intended.	**Experiences of workplaces** Explore how coding is used in game development studios, software companies, robotics laboratories and technology businesses. Use examples of games and applications to show how code controls actions, responds to inputs and keeps track of scores. **Encounters with further and higher education** Introduce pupils to opportunities to develop computing skills through school, coding clubs, college courses, university degrees and technical training in programming, software development and digital technology.