Broadway Junior School

Subject: Computing Topic: Coding Year: 3 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, Bug, Button, Click Events, Code, Collision Detection Event, Command, Debug/Debugging, Degrees, Event, Flowchart, Implement, Input, Interval, Nest, Object, Predict, Properties, Repeat, Right-Angle, Run, Scene, Sequence, Test, Timer, Turtle Object.

Knowledge	Skills
• To understand what a flowchart is and how flowcharts are used in computer programming. • To understand that there are different types of timers and select the right type for purpose. • To understand how to use the repeat command. • To understand the importance of nesting. • To design and create an interactive scene.	• Children can read and explain a flowchart. • Children can use a flowchart to create a computer program. • Children can create a computer program that uses click events and timers. • Children can create a program that uses a timer-after command. • Children can create a program that uses a timer-every command. • Children understand there can be different ways to solve a problem. • Children understand how the turtle object moves. • Children can use the repeat command with an object. • Children can create a computer program that includes use of the repeat command. • Children can create computer programs using prior knowledge. • Children can run, test and debug their programs.

Broadway Junior School

Subject: Computing Topic: Coding Year: 3 Term: Spring 1

	• Children can consider nesting when debugging their programs • Children can use the attributes (properties) table to set the attributes of objects. • Children can plan their scene and code before they create their program. • Children can confidently make several different things happen in a program.

Learning Objectives/End Points					
To understand that computer programs work by following algorithms and to use simple planning tools. Lesson 1	To understand how objects, actions and events work together in a program. Lesson 2	To understand the purpose of an output and to be able to 'read' code to find out what it does. Lesson 3	To use repeat and timer commands to control how programs run. Lesson 4	To design a program. Lesson 5	To use design documentation to code a program. Lesson 6

Prior and Future Learning Links:
• Unit 2.1 (Coding): Algorithms, Collision detection, Timers, Objects, Buttons, Debugging. Unit 4.1 (Coding): Code-test-debug process, IF statements, Repeat until and IF/ELSE statements, Number variables. Unit 4.5 (Logo): Text-based coding, Utilise understanding of coding structures.

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 examples of careers that use coding, such as software developers, game designers, robotics engineers and web developers. Explain that these jobs involve designing, testing and improving digital solutions. **Addressing the needs of each pupil** Provide accessible coding activities with appropriate scaffolds, visual flowcharts, modelling and opportunities for extension so all pupils can develop confidence and problem-solving skills. **Linking curriculum learning to careers** Link algorithms, sequences, events, timers, repeat commands, nesting and debugging to the work of programmers and game developers. Highlight how planning, testing and correcting errors help people create programs that work as intended. **Encounters with employers and employees** Where possible, invite a parent, school staff member or visitor who works with technology to discuss how they use problem-solving, programming or digital tools in their job. **Experiences of workplaces**

Broadway Junior School

Subject: Computing Topic: Coding Year: 3 Term: Spring 1

Transferable skills Logical thinking: Understanding algorithms, sequences and flowcharts. Problem-solving: Finding different ways to achieve a programming outcome. Creativity: Designing and coding interactive scenes. Resilience: Testing, debugging and improving programs when errors occur. Planning and organisation: Using design documentation before writing code. Attention to detail: Checking commands, timers, events and nesting. Evaluation: Predicting what code will do and testing whether it works as intended.	Explore how code is used in real-world settings, such as game studios, robotics laboratories, animation companies and businesses that develop websites or interactive applications. **Encounters with further and higher education** Introduce pupils to the idea that computing skills can be developed further through school, computing clubs, college courses, university degrees and technical training in programming and digital technology.