Broadway Junior School

Subject: Computing Topic: Coding Year: 6 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
Abstraction Algorithm Decomposition Datatype Event Action Function repeatedly. Initialize Nesting Coordinates Flowchart algorithm. 'If' statement 'If/Else' statement Object Selection Repeat forever. Tag Attribute Variable program is running. Random String Physical System Simplify Simulation

Knowledge	Skills
• To understand that computer programs work by following algorithms and to use simple planning tools. • To build structured programs using events and outputs and debug using logical reasoning. • To use selection and co-ordinates to control program behaviour. • To use repeat and timer commands to control how programs run. • To understand what a variable is in programming and how datatype affect how variables are used. • To understand what a function is and how functions work in code.	• I can explain that an algorithm is a step-by-step set of instructions. • I can arrange code blocks to create a simple program. • I can read and explain a simple flowchart. • I can run and test my code. • I can explain how events trigger actions. • I can combine object, action, event and output blocks. • I can read code line-by-line to predict outcomes. • I can debug code by comparing it to an algorithm. • I can use an 'if' statement to control behaviour. • I can extend this using 'if/else'. • I can explain how X and Y co-ordinates position objects.

Broadway Junior School

Subject: Computing Topic: Coding Year: 6 Term: Spring 1

	• I can combine selection with co-ordinates. • I can use a repeat command to make actions happen multiple times. • I can explain the difference between timer-after and timer-every commands. • I can explain what nesting means. • I can explain what a variable is in programming. • I can choose the appropriate number or string datatype when creating variables. • I can create and use variables when programming. • I can create functions in code. • I can use functions to make my programming more efficient. • I can use the friction attribute within my code.

	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 and how datatype affect how variables are used. Lesson 5	To understand what a function is and how functions work in code. Lesson 6

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. Transferable skills such Computational thinking: Breaking complex problems into smaller, manageable steps through decomposition and abstraction. Logical reasoning: Predicting how programs will behave and explaining the purpose of algorithms and flowcharts.	**Learning from career and labour market information** Explore the demand for programmers, software developers, cybersecurity specialists and engineers, including how coding supports industries across the North East and beyond. **Addressing the needs of each pupil** Provide scaffolded coding tasks, visual flowcharts, worked examples and appropriately challenging programming problems so all pupils can develop confidence and progress. **Linking curriculum learning to careers** Apply algorithms, decomposition, abstraction, selection, variables, functions and debugging to develop programs, reflecting the skills used by professionals who design and maintain software. **Encounters with employers and employees** Explore how software developers, game designers and engineers use algorithms and code in their work. Where possible, arrange a virtual or in-person encounter with a professional working in technology. **Experiences of workplaces** Use realistic programming challenges, such as creating a simple game, simulating a physical system or designing an interactive program, to introduce pupils to workplace coding tasks**.** **Encounters with further and higher education** Introduce pupils to computing and programming pathways through secondary school, college, apprenticeships, university courses and technical qualifications.

Broadway Junior School

Subject: Computing Topic: Coding Year: 6 Term: Spring 1

Problem-solving: Identifying errors, debugging code and refining solutions through systematic testing. Creativity and innovation: Designing programs using events, objects, actions and outputs to achieve a specific purpose. Attention to detail: Using coordinates, selection statements, variables, data types and functions accurately. Organisation and planning: Structuring programs and using functions to make code more efficient and reusable. Resilience: Testing, improving and debugging programs when they do not behave as expected. Communication and collaboration: Explaining algorithms, discussing solutions and sharing approaches to programming challenges.	