ST. DUNSTAN'S

Computing Curriculum Planning

Computer Science (Computing sessions)

Children at St Dunstan’s are living in a world where technology has completely changed virtually every aspect of our society over the past few decades, from the way we work to the way we socialise and everything in between. We are teaching computer science to prepare them for living and working in a digital society where technology is rapidly developing. 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. At 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 Science

EYFS	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
1. Beebots. Explore using the Beebots.	**1. Algorithm direction cards to build Lego tower and program Bee bots.** Algorithms: Building Lego Towers and using BeeBots - Algorithm is a set of instructions. - Use the word debug. - Use logical reasoning to predict the behaviour of simple programs.	**1. Light bot, progress to introduction to scratch** (change a background and select a sprite) - Simple programming -Understand that fixing means debugging - Order of instructions - Basic Introduction to Scratch. Introduce sequencing.	**1. Scratch** a. Create a sprite b. Animate name or dance party c. Speech bubbles - Sequencing - Start to identify what to debug. -Repeat commands	**1. Probots** - evaluate and apply - follow logical sequence - consider potential errors - Scratch pen drawing to test program **2. Hour of code Minecraft** - Debugging, start to consider potential errors. - Sequence and repetition under instruction.	**1. Scratch Create a game.** (Clicker game, chase game, pong game) - Design and create own program. - Sequence and repetition independently. - Selection, if … then... - Debugging, start to pre-empt potential errors. **2. Microbit** Sequence and repetition independently. - Debugging, start to pre-empt potential errors.	**1. Lego Mindstorm** - Design and create own program confidently. - Control models and characters. - Testing and debugging independently.