Curriculum Overview
Subject: Computing
2017-18

Vision for Computing: At the Westminster School we believe that 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 our 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.

		Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
KS2		Create and debug simple programs, using Bee Bot toy, Bee Bot on Ipads, Daisy Dino	Use technology purposefully to create, organise, store, manipulate and retrieve digital content. Use iPad Book Creator and Our Story, Create a presentation about myself	Understanding algorithms, Barefoot SEN programme of study	Understanding algorithms, Barefoot SEN programme of study	Logical reasoning, Lego Duplo Train (iPad), Pro-Bot, Probotix	Create and debug simple programs, Scratch Jr E-Safety
		System Life Cycle	**Inside a Computer**	**Computer Logic**	**Programming**	**Visual Programming**	**E Safety**
Yr 7/8		Bamzooki, Purple Mash, Google SketchUp, Understand the Design, Creation, Testing stanges	Take a PC apart, introduction to computer hardware	Light Bot, Smart Kit, Bee Bot Learn how computers use logic	Hour of Code, Just 2 Easy, Begin to experience computer programming	Scratch Jr, Programming with symbols	Understand the dangers of the digital world
Year 9		Programming and Hardware BBC Micro Bit	The Human , Computer Interface CBBC Technobabble Game Maker	Games Programming Kodu	Development Environments and Programming, Star Logo TNG	Programming with a Sprit and Variables, Scratch	Controlling a Device in the real world, Pro-Bot, Probotix

	Strad 1 Computer hardware and software Computer memory, storage and moral, legal and environmental concerns	Strad 1 Computer hardware and software Computer memory, storage and moral, legal and environmental concerns	Strand 2 Computational logic and algorithms	Strand 2 Computational logic and algorithms	Strand 3 Programming techniques and data representation	Strand 3 Programming techniques and data representation
Year 10	Strad 1 Computer hardware and software Computer memory, storage and moral, legal and environmental concerns	Strad 1 Computer hardware and software Computer memory, storage and moral, legal and environmental concerns	Strand 2 Computational logic and algorithms	Strand 2 Computational logic and algorithms	Strand 3 Programming techniques and data representation	Strand 3 Programming techniques and data representation
Year 11	Strad 1 Computer hardware and software Computer memory, storage and moral, legal and environmental concerns	Strad 1 Computer hardware and software Computer memory, storage and moral, legal and environmental concerns	Strand 2 Computational logic and algorithms	Strand 2 Computational logic and algorithms	Strand 3 Programming techniques and data representation	Strand 3 Programming techniques and data representation

Over KS3 & 4 pupils at The Westminster will be taught to:

- design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems
- understand several key algorithms that reflect computational thinking [for example, ones for sorting and searching]; use logical reasoning to compare the utility of alternative algorithms for the same problem
- use 2 or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions
- understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in circuits and programming; understand how numbers can be represented in binary, and be able to carry out simple operations on binary numbers [for example, binary addition, and conversion between binary and decimal]
- understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems
- understand how instructions are stored and executed within a computer system; understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits
- undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users
- create, reuse, revise and repurpose digital artefacts for a given audience, with attention to trustworthiness, design and usability
- understand a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct, and know how to report concerns