Computer Science

At St. Saviour's & St. Olave's it is imperative that we develop students into active users and creators of technology, understanding and questioning their role within it. Students critically reflect on the developments of Computer Science and its impact on ethics, culture and the environment and are given the space to examine within lesson. Technology should enable our students to flourish in all aspects of their life so it is important that they become confident, critical and informed. Our curriculum allows students to develop their programming, logical thinking, cyber and e-safety and digital literacy skills whilst honing their use of popular software packages used in the professional world. As Computer Science is ever changing and developing daily, it is important for us to regularly update and improve our curriculum based on feedback from students, staff and wider industry professionals. Furthermore, it is imperative that we offer a fully inclusive and transformative curriculum which enables all students to flourish and provides a range of opportunities outside of school to help enrich and apply their skills.

KS3 Curriculum for Computer Science

At KS3, students develop their understanding and use of networks, data representation, programming, cybersecurity and e-safety. They also enhance their use of Office applications through project-based units where they work collaboratively and respectfully using online tools. Students develop a systematic approach to problem solving through abstraction and decomposition when developing coded solutions. They also learn how to make effective progress through applying exam technique and understanding how to provide and act on constructive feedback. These key skills provide a good basis for GCSE Computer Science but equally, can be applied to all subjects across the curriculum.

	Half-term 1	**Half-term 2**	**Half-term 3**	**Half-term 4**	**Half-term 5**	**Half-term 6**
Year 7	Introduction to the network and Google Classroom Email, feedback and presentation skills, and how to work collaboratively.	E Safety How to be an upstander, how to detect and avoid cyber threats and how to prevent them.	Inside a computer Learn about the history of Computer Science, "speak" binary like a computer, build a paper model of a laptop and differentiate between input and output devices.	Introduction to programming with Scratch Create animations and games using Scratch. Apply sequence and selection concepts when programming.	Spreadsheet modelling Use a spreadsheet to help organise, calculate and format data. Use a spreadsheet model to help plan a birthday party.	End of year revision Pop Art using Google Draw Create a Pop Art masterpiece using a vector-based graphics program.
Year 8	Cryptography and encryption	Interactive "Choose Your Own Adventure" stories	Spreadsheet modelling	Introduction to programming using Python (inputs, sequence, selection)	Data Representation introduction (binary/denary conversion, binary addition, character sets, images)	End of year revision App development project
Year 9	Data Representation Part 2 (hexadecimal, binary subtraction and addition, arithmetic shift, logic gates)	Legislation and ethics	Developing AI programs using Python (turtle, GUI, inputs, sequence, selection, iteration)	Cybersecurity	Networks and topologies	End of year revision Python programming project

KS4 Curriculum for Computer Science

Course: GCSE Computer Science

Exam board: Eduqas

Students continue to explore aspects of Computer Science theory and develop and apply critical thinking, analysis and problem-solving skills in a creative and innovative way. There are 2 examined units which are equally weighed and there is no coursework. In the first exam, content is assessed in a theoretical way whereas in the second exam it is assessed on-screen through the development of a program in Python.

	Half-term 1	**Half-term 2**	**Half-term 3**	**Half-term 4**	**Half-term 5**	**Half-term 6**
Year 10	Hardware: CPU, primary and secondary storage, embedded systems. Data Representation: data types, representation of numbers	Operating Systems: Purpose and management, utility software Data representation: representation of graphics and sound, storage requirements, compression Logic operations: Boolean algebra, algebra simplification	Algorithms and constructs: interpreting and constructing algorithms, searching and sorting Paper 2 programming Program construction	Networking and cybersecurity: topologies, OSI model, importance of creating standards, network protocols, hardware, standards, Paper 2 programming	Networks: validation and verification Paper 2 programming	Software development End of year assessment revision Paper 2 programming
Year 11	Impacts, legislation, environment Networking and cybersecurity	Software development: IDE tools, HLL and LLL Program construction: translators, programming errors Python programming 1 lesson per week	Data structures and file design Python programming 2 lessons per week	Algorithmic thinking Exam revision Python programming 2-3 lessons per week	Paper 1 and Paper 2 practice Exam revision	

KS5 Curriculum for Computer Science

In our sixth form, our students analyse problems in computational terms, employ critical thinking and creative problem solving whilst developing their resilience and applying mathematical skills. The course has 3 units including 2 exams and a programming project in Year 13. This provides excellent preparation for further study or employment in Computer Science, however, the skills learned are highly transferrable across other STEM subjects, as well as Art and Humanities.

	Half-term 1	**Half-term 2**	**Half-term 3**	**Half-term 4**	**Half-term 5**	**Half-term 6**
Year 12	Structure and function of the processor Types of processors Input, output, storage Databases Dedicated programming lessons throughout Year 12	Systems software Application generation Software development	Type of programming language Compression, encryption and hashing	Networks Web technologies	Data types Data structures	Boolean algebra Computer-related legislation Ethical, moral and cultural issues Analysis of NEA project
Year 13	Thinking abstractly, ahead, procedurally, logically, concurrently Boolean algebra Programming techniques Ethical, moral and cultural issues. Design of NEA project	Algorithms Mock exam preparation Development and testing of NEA project	Computational methods Paper 1 and 2 practice Testing and evaluation of NEA project	Revision and Paper 1 and 2 practice	Revision	