ST GEORGE'S SCHOOL
A CHURCH OF ENGLAND ACADEMY

Computer Science Curriculum

Computer Science Curriculum Philosophy

The Computer Science Department at St. George's is committed to embedding and highlighting the central Christian values of dignity, hope, community, wisdom, humility, and kindness in our curriculum. We centre our teaching on recognising the inherent value and worth of each individual, aiming to achieve this by providing a high-quality computing education that equips pupils to use computational thinking and creativity to understand and change the world.

To instil hope in our pupils, we explore technological advancements and pioneers who offer inspiration and challenge pupils' perspectives, addressing possible misconceptions about the digital world. We encourage thought-provoking discussions and collaborative projects to foster personal growth and a sense of community within the classroom.

Pupils reflect upon the values of wisdom and humility to broaden their perspectives and challenge their own beliefs about technology and its impact on society. We strive to develop critical thinking and problem-solving skills, deepening pupils' understanding of the complexities of computer science, its ethical implications, and its influence on human societies, cultures, and thought.

As a discipline, computer science inspires curiosity and fascination with how technology shapes our world and the endless possibilities it offers for the future. Our curriculum covers foundational computing principles as outlined in the national curriculum, ensuring pupils understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms, and data representation. We aim to enhance pupils' clarity of thought and self-expression through the development of powerful knowledge and technical vocabulary. It is our intention that our pupils are not constrained by any barriers—social, cultural, or technological.

We challenge our pupils to think as computer scientists. Therefore, we have developed five departmental 'principles' that the teaching team adheres to:

1. **Provide Direct Instruction and Modelling**: We are devoted to teaching foundational computing concepts through clear, direct instruction and modelling. By demonstrating processes and solutions, we equip pupils with the necessary schemas to understand and apply their knowledge effectively.
2. **Develop Computational Thinking**: We focus on building pupils' abilities to think algorithmically and solve complex problems using abstraction, logic, algorithms, and data representation.
3. **Facilitate Guided Practice and Application**: After modelling, we provide opportunities for pupils to apply their knowledge through guided practice, allowing them to solve problems using the base knowledge as a schema.
4. **Contextualise Learning**: We support pupils in connecting theoretical concepts to real-world applications to develop a practical understanding of

how digital systems work and how to put this knowledge to use through programming.

5. **Utilise a Concept-Driven Curriculum**: We employ a curriculum centred on key computing concepts to build deep and transferable understanding, ensuring pupils can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems.

We utilise models of progression to empower pupils to develop an ever-improving appreciation of the 'big ideas' in computer science and the forces that drive technological innovation. Our pupils develop an understanding of substantive concepts (e.g., algorithms, programming languages, data structures) and disciplinary concepts (e.g., abstraction, decomposition, pattern recognition) so that they might produce increasingly sophisticated solutions.

We promote a love of learning. We guide our pupils towards developing an increasing level of independence in their use of analytical and coding skills and in making judgements about technological solutions. We lead our pupils to develop clarity of thought and expression, both in written and programming languages. We promote Christian and British values so that our pupils appreciate the qualities that make individuals unique. We nurture a strong work ethic and celebrate success. We teach our pupils to recognise that the values, powerful knowledge, and skills they develop in computer science will support the growth of their self-confidence and personal resilience. These values will be important in helping our pupils to succeed academically, as family members, citizens, and future leaders and colleagues.

In order to achieve an increasingly sophisticated understanding of computer science, topics have been intelligently sequenced based on the following rationale:

- **Foundation Building Through Direct Instruction**: We begin with foundational concepts such as understanding the principles of information and computation, how digital systems work, and basic programming skills. These are taught through direct instruction and modelling to ensure all pupils have a solid base upon which to build more complex ideas.
- **Progressive Complexity with Guided Practice**: Topics are arranged to gradually increase in complexity. Pupils apply their knowledge through guided practice, allowing them to make connections between concepts and develop problem-solving skills using their established schemas.
- **Interdisciplinary Links**: We incorporate elements of mathematics, science, and design and technology to provide a holistic understanding of how computer science interacts with other disciplines. Pupils learn to understand key algorithms, use logical reasoning, and apply mathematical concepts such as binary numbers and Boolean logic.
- **Real-World Applications**: We sequence topics to align with current technological trends and societal needs, making learning relevant and engaging. Pupils undertake projects that involve selecting, using, and combining multiple applications to achieve challenging goals, including collecting and analysing data and meeting the needs of known users.

To maximise the impact of our curriculum, the following elements have been selected as 'building blocks':

- **Direct Instruction and Modelling**: Clear teaching of concepts and procedures, providing pupils with the necessary knowledge and skills. Teachers model problem-solving processes, programming techniques, and computational methods.
- **Structured Practice and Application**: Pupils engage in structured activities that reinforce learning. Guided practice transitions to independent practice as pupils become more proficient.
- **Computational Thinking Skills**: Developing problem-solving strategies, logical reasoning, and algorithmic thinking. Pupils analyse problems in computational terms and gain practical experience in writing computer programs.
- **Understanding Digital Systems**: Knowledge of hardware and software components that make up computer systems, how they communicate, and how instructions are stored and executed.
- **Digital Literacy and Ethical Use**: Ensuring pupils become responsible, competent, confident, and creative users of information and communication technology. Pupils learn to use technology safely, respectfully, responsibly, and securely.

The computer science curriculum addresses social disadvantage by bridging gaps in pupils' knowledge and skills:

- **Equitable Access to Knowledge**: Through direct instruction, all pupils receive the same high-quality teaching, ensuring that everyone has access to the foundational knowledge required to succeed.
- **Supportive Learning Environment**: We provide additional support where needed, including resources and interventions for pupils who may require extra help in understanding key concepts.
- **Inclusivity Through Representation**: We highlight contributions from diverse cultures and communities within the field of computer science, promoting respect for diversity and encouraging all pupils to see themselves as potential contributors to the field.

We believe that computer science contributes to the personal development of pupils at St. George's:

- **Building Confidence Through Mastery**: By providing clear instruction and opportunities for success, pupils develop confidence in their abilities, reflecting the Christian value of hope.
- **Developing Resilience**: Pupils learn to persevere through challenging tasks, such as debugging code, which builds personal resilience.
- **Promoting Ethical Understanding**: Discussions on topics like data privacy and digital citizenship help pupils consider the moral implications of technology, fostering spiritual and moral development.
- **Encouraging Collaboration and Respect**: Group projects and peer learning foster teamwork, communication skills, and respect for others' ideas, enhancing social development and a sense of community.

At both Key Stage 3 and Key Stage 4, our belief is that homework should reinforce learning and promote independent study skills:

- **Reinforcement of Taught Concepts**: Homework assignments focus on practising and applying concepts that have been directly taught and modelled in class.
- **Interleaved Practice**: Assignments are designed to revisit and reinforce previous learning, aiding in knowledge retention and schema development.
- **Progressive Independence**: As pupils' proficiency improves, homework tasks encourage greater independence, preparing them for more complex problem-solving.

Opportunities are built in to make links to the world of work to enhance the careers, advice, and guidance that pupils are exposed to:

- **Careers Spotlight**: Each topic has a 'careers spotlight', where pupils explore professions linked to that particular unit of work. Information about qualifications needed, salaries, and career progression is provided.
- **Skills for the Workplace**: Emphasis on problem-solving, critical thinking, planning, and organisation prepares pupils for a variety of careers. Pupils learn to present solutions, consider the ideas of others, and work effectively in teams.
- **Industry Connections**: Through relationships with local technology companies and universities, pupils experience real-life applications of computer science and the variety of careers available in the field.

We ensure pupils are well prepared to be successful in GCSE examinations and beyond:

- **Depth of Knowledge**: While adhering to the national curriculum, we extend learning by exploring topics in greater depth, ensuring pupils have a robust understanding of key concepts.
- **Encouraging Further Study**: We promote higher education pathways in computer science and related fields, inspiring pupils to pursue their passions.
- **Resource Provision**: We provide a comprehensive library of computing resources, including books, online platforms, and software tools, to support independent learning.

In addition to the references already made to the spiritual, moral, social, and cultural development of the pupils, the Computer Science curriculum embeds the following links:

- **Spiritual Development**: Pupils are encouraged to reflect on the impact of technology on society and consider the bigger questions about the role of technology in improving lives.
- **Moral Development**: The curriculum addresses ethical issues related to technology use, including privacy, intellectual property, and the digital divide.
- **Social Development**: Collaborative learning and group projects help pupils develop social skills, including communication, teamwork, and respect for diversity.
- **Cultural Development**: Pupils explore how technology influences culture and how different cultures contribute to technological advancements.

Christian and British Values

- **Democracy**: Pupils learn to work collaboratively, make group decisions, and respect the views of others.
- **Rule of Law**: Understanding and adhering to laws related to technology use, such as data protection and copyright laws.
- **Individual Liberty**: Encouraging pupils to express themselves through technology, while understanding the responsibilities that come with digital freedoms.
- **Mutual Respect and Tolerance**: Fostering an inclusive environment where all contributions are valued, and differences are respected.

Our curriculum is systematically reviewed and revised to ensure we meet the specific needs and requirements of our pupils, aligning with the national curriculum guidelines. We stay abreast of recent developments in pedagogy and technological advancements to inform necessary changes to our computer science curriculum.

In all aspects, we strive to provide an inclusive, engaging, and challenging educational experience that equips pupils with the knowledge, skills, and values necessary to thrive in an increasingly digital world.

Curriculum Sequencing

All children are entitled to a curriculum and to the powerful knowledge that will open doors and maximise their life chances. Below is a high-level overview of the critical knowledge children will learn in this particular subject, at each key stage from Year 10 through to Year 11, in order to equip pupils with the cultural capital they need to succeed in life. The curriculum is planned vertically and horizontally giving thought to the optimum knowledge sequence for building secure schema.

Unit Sequencing			
Year 8	**Progress Period**	**6 weeks**	**Unit 1: Computer Systems** In Unit 1, pupils delve into the inner workings of computer systems, exploring key hardware components such as the CPU and GPU, alongside primary and secondary storage. They also examine the roles of operating systems and application software, gaining a comprehensive understanding of how technology powers modern computing.
		6 weeks	**Unit 2: Bitmap and Vector Images** In Unit 2, pupils explore the differences between bitmap and vector images. They learn how bitmaps use pixels, with colour depth and resolution linked to binary, while vectors use scalable shapes defined mathematically. Pupils develop editing skills in both bitmap and vector software to create digital graphics.
		2 weeks	**Unit 3: Boolean Logic** In Unit 3, pupils explore Boolean logic, focusing on the operators AND, OR, and NOT. They learn how these logical operators are used in decision-making processes and apply them in preparation for their advanced block programming unit, where Boolean logic enhances their coding capabilities.
		4 weeks	**Unit 4: Block Programming Advanced** In Unit 3, pupils advance their programming skills using Scratch by introducing subroutines with custom blocks. They learn to create modular programs by breaking tasks into smaller, reusable parts, explore advanced conditionals, and use loops effectively. Pupils also practise debugging, enabling them to build efficient and organised projects.

Year 9	**Progress Period**	**6 weeks**	**Unit 1: Cybersecurity** In Unit 1, pupils delve into the world of cybersecurity. They explore key concepts such as data protection, phishing, malware, hacking, and safe online practices. Pupils also examine the ethical and legal implications of cyber activities, including privacy rights, responsible online behaviour, and the consequences of illegal hacking or data breaches. They learn to identify potential threats, understand the importance of secure systems, and develop strategies to protect personal and organisational data.
		6 weeks	**Unit 2: Multimedia Production** In Unit 2, pupils explore how audio and video are captured, stored, and processed, linking this to computing concepts as data representation as well as sampling, bit depth, and file compression. They apply this knowledge in a creative project, learning to edit and produce multimedia content using industry-relevant audio and video editing tools.
		6 weeks	**Unit 3: Text Based Programming** In Unit 3, pupils are introduced to text-based programming using Python. They build on concepts learned in block-based programming, such as sequencing, selection, and iteration, while exploring new elements like string handling and lists. This unit develops their coding skills and prepares them for more advanced programming tasks at KS4.

Key Stage 4 Computer Science

Curriculum Sequencing

At Key Stage 4, pupils who opt for Computer Science study the OCR GCSE specification. The curriculum sequencing below has been developed with reference to the OCR GCSE specification - https://www.ocr.org.uk/images/558027-specification-gcse-computer-science-j277.pdf

Key Stage 4		Sequencing	
		Paper 1	**Paper 2**
Year 10	Term 1	**1.1-Systems architecture** • **1.1.1 Architecture of the CPU** • **1.1.2 CPU performance** • **1.1.3 Embedded systems** **1.2- Memory and storage** • **1.2.1 Primary storage (memory)** • **1.2.2 Secondary storage** • **1.2.3 Units**	**2.2-Programming fundamentals** • **2.2.1 Programming fundamentals** • **2.2.2 Data types** • **2.2.3 Additional programming techniques** • **2.3-Producing robust programs** • **2.3.1 Defensive design** • **2.3.2 Testing**
	Term 2	• **1.2.4 Data storage (Numbers, Characters, Images, Sound)** • **1.2.5 Compression** **1.5-Systems software** • **1.5.1 Operating systems** • **1.5.2 Utility software**	**2.1 - Algorithms** • **2.1.1 Computational thinking** • **2.1.2 Designing, creating and refining algorithms** • **2.1.3 Searching and sorting algorithms** **2.4-Boolean logic**
	Term 3	**1.3-Computer networks, connections and protocols** • **1.3.1 Networks and topologies** • **1.3.2 Wired and wireless networks, protocols and layers** **1.4- Network security** • **1.4.1 Threats to computer systems and networks**	**2.5-Programming languages and Integrated Development Environments** • **2.5.1 Languages**

Key Stage 4		Sequencing	
		Paper 1	**Paper 2**
		• 1.4.2 Identifying and preventing vulnerabilities	• 2.5.2 The Integrated Development Environment (IDE) Practical Programming skills project Pseudocode Practice
Year 11	Term 1	1.6-Ethical, legal, cultural and environmental impacts of digital technology • 1.6.1 Ethical, legal, cultural and environmental impact	Practical Programming skills project Pseudocode Practice
	Term 2	Extended Revision • Reteaching and retrieval practice of Paper 1 content	Extended Revision • Reteaching and retrieval practice of Paper 2 content
	Term 3	Extended Revision • Reteaching and retrieval practice of Paper 1 content • Past papers, targeted intervention, and exam technique.	Extended Revision • Reteaching and retrieval practice of Paper 1 content • Past papers, targeted intervention, and exam technique.