



St George's School

Computer Science

KS4 Curriculum

PRIOR KNOWLEDGE <i>Knowledge and skills developed in KS3</i>	Computing specific knowledge as detailed in our KS3 curriculum maps. Skills developed: <ul style="list-style-type: none">- Knowledge of computer Systems- Algorithms and programming- Computational thinking- Software development cycle
COURSE DELIVERY & STRUCTURE <i>How the curriculum is delivered</i>	Lessons: In year 10 and 11, 5 one hour lessons every two weeks. Grouping: There are no sets. Structure: Year 9 is an Entry Level Certificate course from OCR. The OCR GCSE builds on this foundation course. The course is split into two components, 'Computer Systems' and 'Computational Thinking, Algorithms and Programming'. These components are broken down further into a series of units. Each component comprises 50% of the course. There is no programming project component. Prep: Students normally one receive one prep a week. .
QUALIFICATION <i>Exam Board, aim and objectives</i>	GCSE (9-1) in Computer Science OCR J277 Qualification aims and objectives: <ul style="list-style-type: none">➢ understand and apply the fundamental principles and concepts of Computer Science. This includes abstraction, decomposition, logic, algorithms, and data representation.➢ analyse problems in computational terms through practical experience of solving such problems, including designing, writing and debugging programs.➢ think creatively, innovatively, analytically, logically and critically.➢ understand the components that make up digital systems, and how they communicate with one another and with other systems.➢ understand the impacts of digital technology to the individual and to wider society
ASSESSMENT <i>Internal monitoring and final assessment</i>	Internal Assessment: Prep assignments are monitored for completion and accuracy. Check-point assessments take place after every topic. Students complete mock exams in Year 10 and Year 11. Final assessment: 2 equally-weighted written examination papers, each of which is 1 hour and 30 minutes long. Both exams are non-calculator.
BREADTH <i>Opportunities, trips, wider reading, cultural capital</i>	Isaac Computer Science https://isaaccomputerscience.org/ Bebas Challenge https://www.bebas.uk/

SUBJECT KNOWLEDGE <i>Overview of topics</i>		SKILLS & STRATEGIES <i>Procedural knowledge</i>
Autumn Y10	Binary Units of Data Storage. Boolean Logic Character Sets Sound Images Compression Legislation Technology Issues	Why computers use binary; binary shifting; binary arithmetic; binary conversions to and from denary and hexadecimal numbers. Bit; nibble; byte; kilobyte; megabytes; gigabyte; terabyte; petabyte. Logic gates (AND, OR & NOT); combining logic gates; truth tables; Boolean algebra. Binary representation; ASCII; Unicode. Binary representation; digital sound; sampling; storage; playback quality. Binary representation; metadata; colour depth; resolution; image quality; storage. Lossy; lossless. The Data Protection Act; Computer Misuse Act; Copyright Designs and Patents Act; open source and proprietary software. Ethical issues; cultural issues; environmental issues; privacy issues.
Spring Y10	Programming Fundamentals Programming Constructs Operators Subroutines Data Storage Common Algorithms	Variables; constants; data types; input; output; typecasting; arrays. Sequence; selection; iteration. Arithmetic operators; comparison operators; boolean operators; string manipulation. Procedures; functions; parameters; arguments; local variables; global variables. Records; using files (opening, closing, reading, writing); Structured Query Language (SQL); basic SQL commands. Linear search; binary search; bubble sort; merge sort; insertion sort.
Summer Y10	Defensive Design Testing Integrated Development Environments Syntax and Logic Errors Programming Languages	Anticipating misuse; authentication; input validation; maintainability. Iterative testing; final testing; beta testing; appropriate test data. Common tools; editors; error diagnostics; run -time environment; debugging. Identifying and correcting program errors. High level languages; low level languages; translators; compilers; interpreters; machine code.

Autumn Y11	Operating Systems Utility Software Computational Thinking Algorithms Primary Memory Secondary Storage	User interfaces; memory management and multitasking; peripherals; user management; file management. Encryption software; defragmentation; data compression; backups. Abstraction; decomposition; algorithmic thinking. Inputs, processes and outputs; pseudocode; flowcharts; trace tables. RAM; ROM; virtual memory. Optical; magnetic; solid -state; advantages and disadvantages of each technology.
Spring Y11	CPU Architecture CPU Performance Embedded Systems Types of Network Network Hardware Network Topologies. The Internet Wired and Wireless Networks Network Protocols Network Layers Network Threats Network Threat Prevention	Fetch-execute cycle; ALU; CU; cache; registers; MAR; MDR; program counter; accumulator; Von Neumann architecture. Clock speed; cache size; number of cores; RAM. Purpose and examples. LAN; WAN; client-server; peer to peer; network performance. Wireless access points; routers; switches; NICs; transmission media. Star; mesh. DNS; hosting; the cloud. Modes of connection; encryption; IP addressing; MAC addressing; packet switching. TCP/IP; HTTP; HTTPS; FTP; POP; SMTP; IMAP. Advantages; development; maintenance. Malware; social engineering; brute-force attacks; denial of service attacks; data interception and theft; SQL injection. Prevention methods; penetration testing; anti -malware; firewalls; user access levels; passwords; encryption; physical security.
Summer Y11	Revision, consolidation and exam preparation	Spaced repetition; active recall; past papers.