St George’s School
Computer Science
KS4 Curriculum

PRIOR KNOWLEDGE *Knowledge and skills developed in KS3*	Computing specific knowledge as detailed in our KS3 curriculum maps. Skills developed: - Knowledge of computer Systems - Algorithms and programming - Computational thinking - Software development cycle
COURSE DELIVERY & STRUCTURE *How the curriculum is delivered*	**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 *Exam Board, aim and objectives*	GCSE (9-1) in Computer Science OCR J277 Qualification aims and objectives: ➢ 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 *Internal monitoring and final assessment*	**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 *Opportunities, trips, wider reading, cultural capital*	Isaac Computer Science https://isaaccomputerscience.org/ Bebras Challenge https://www.bebras.uk/

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

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