All Saints' Academy Computer Science Department KS4 Curriculum Overview - September 2023_24

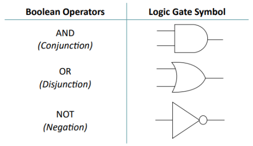
Aims and Learning Outcomes

OCR's GCSE (9–1) in Computer Science will encourage students to:

Understand and apply the fundamental principles and concepts of Computer Science, including 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
Apply mathematical skills relevant to Computer Science.

Assessment Overview	
Written paper: 1 hour and 30 minutes **50%** of total GCSE 80 marks This is a non-calculator paper.	**Written paper: 1 hour and 30 minutes** **50%** of total GCSE 80 marks This is a non-calculator paper. This paper has two sections: Section A and Section B. Students must answer both sections

All Saints’ Academy Computer Science Department KS4 Curriculum Overview - September 2023_24

Cycle	Year 11		Enrichment
Careers	Software Developer/Engineer, Data Scientist, Cybersecurity Analyst, Web Developer, Systems Analyst, Network Administrator, Artificial Intelligence (AI) Engineer, UX/UI Designer, IT Project Manager, Database Administrator, Computer Programmer, IT Consultant & Game Developer		
1	**Algorithms** Principles of computational thinking: o Abstraction o Decomposition o Algorithmic thinking Identify the inputs, processes, and outputs for a problem, Structure diagrams, Create, interpret, correct, complete, and refine algorithms using: • Pseudocode, Flowcharts, Reference language/high-level programming language Identify common errors ¨ Trace tables Standard searching algorithms: • Binary search o Linear search • Standard sorting algorithms: ○ Bubble sort, Merge sort, Insertion sort **Programming Fundamentals** The use of variables, constants, operators, inputs, outputs and assignments The use of the three basic programming constructs used to control the flow of a program: • Sequence, Selection, Iteration (count- and condition-controlled loops) The common arithmetic operators The common Boolean operators AND, OR and NOT	• Practical use of the data types in a high-level language within the classroom • Ability to choose suitable data types for data in a given scenario • Understand that data types may be temporarily changed through casting, and where this may be useful Practical use of the additional programming techniques in a high-level language within the classroom Ability to manipulate strings, including: • Concatenation, Slicing, Arrays as fixed length or static structures, Use of 2D arrays to emulate database tables of a collection of fields, and records The use of functions The use of procedures, where to use functions and procedures effectively The use of the following within functions and procedures: • local variables/constants • global variables/constants • arrays (passing and returning) SQL commands: • SELECT, FROM, WHERE • Be able to create and use random numbers in a program	Coding Workshops – GCHQ link
Careers	Digital Marketing Specialist, Cloud Architect, Mobile App Developer, Computer Systems Analyst, Information Security Analyst, IT Support Specialist & Technology Teacher/Instructor		
2	**Producing Robust Programs** Defensive design considerations: • Anticipating misuse, Authentication • Input validation • Maintainability: ○ Use of sub programs, Naming convention, Indentation, Commenting The purpose of testing Types of testing: • Iterative, Final/terminal • Identify syntax and logic errors • Selecting and using suitable test data: ○ Normal ○ Boundary ○ Invalid/Erroneous ○ Refining algorithms	**Boolean Logic** Knowledge of the truth tables for each logic gate, Recognition of each gate symbol, understanding of how to create, complete or edit logic diagrams and truth tables for given scenarios. Ability to work with more than one gate in a logic diagram  Boolean Operators | Logic Gate Symbol AND *(Conjunction)* OR *(Disjunction)* NOT *(Negation)* Alternatives Use of other valid notation will be accepted within the examination, e.g. • Using T/F for 1/0, or V for OR, etc **Programming languages and Integrated Development Environments** The differences between high- and low-level programming languages The need for translators The differences, benefits, and drawbacks of using a compiler or an interpreter Common tools and facilities available in an Integrated Development Environment (IDE): • Editors, Error diagnostics, Run-time environment, Translators	Tech Entrepreneurship Challenges Guest Speak: IOActive, Jean Goulding Institute & Cyber First, QA & Sopra Steria
3	**Revision and Exam Preparation** https://teach-ict.com/2016/revision/flashcards_ocr/flashcards_ocr.html https://teach-ict.com/2016/revision/multiple_choice/gcse_ocr_multiple_choice.html https://teach-ict.com/2016/revision/exam_questions_ocr/gcse_ocr_exam_questions_bytopic.html https://teach-ict.com/2016/revision/exam_questions_ocr/gcse_ocr_exam_questions.html https://teach-ict.com/2016/revision/pseudocode_ocr/pseudocode_ocr.html		The use of **SMART REVISE** throughout the year