Computer Science

GCSE

OCR

Course Content

Year 12
Term 1
• Course induction • Prior knowledge checks • Baseline assessment • Input, output and storage devices • Magnetic, flash and optical storage • RAM vs ROM • Virtual storage • Primitive data types • Binary representation • Binary, hexadecimal and denary conversions • Character sets (ASCII, Unicode) • Thinking abstractly • Thinking ahead • Thinking procedurally • Thinking logically • Problem decomposition • Divide and conquer • Structure of the processor • ALU, control unit and registers • Fetch–Decode–Execute cycle • CPU performance factors • Baseline test • Early formative checks
Term 2
• Relational databases • Primary and foreign keys • Entity relationship modelling • Computational problem solving • Strategies for solving problems • Categorising problems • Arrays • Records / tuples • Data organisation • Variables and constants Input/output • Data types • Selection and iteration • Control structures • LAN vs WAN • Domain names and IP addresses • DNS • Summative assessment

Term 3
• Arrays (1D and 2D) • Python implementation of arrays • Bitwise operations • Bit masking • Classes and objects • Attributes and methods • Constructors • Instances • Internet structure • URLs • TCP/IP • Network hardware • Stacks • Queues • Linked lists • Modular programming • Functions and procedures • Parameters and arguments • Global vs local variables • Recursive functions • Tracing recursion • Recursion vs iteration
Term 4
• Algorithm design • Computational methods • Algorithm efficiency • Linear search • Binary search • Bubble sort • Insertion sort • Quick sort • Merge sort • Big O notation • Time complexity types • NEA Project ideas and proposal • NEA Stakeholder analysis • NEA Success criteria • NEA Decomposition • NEA Flowcharts and structure charts
Term 5
• HTML structure • CSS styling • Inline vs external CSS • Web page structure • Client-side processing • Server-side processing • Peer-to-peer models • Operating systems • Memory management • Scheduling algorithms • BIOS • Device drivers • Software development lifecycle models • Agile

- Waterfall
- Rapid application development
- NEA IDE configuration
- NEA Developing solution components
- NEA Testing and iteration
- NEA User feedback and evaluation

Term 6

- Virtual machines
- Installing operating systems in VMs
- Virtual environments
- Network security
- Firewalls and proxies
- Encryption
- Malware (viruses, worms, trojans)

Year 13

Term 1

- Thinking abstractly
- Programming fundamentals
- Problem decomposition
- Inputs, outputs and preconditions for solutions
- Introduction to data structures
- Use of structured data in algorithms
- Boolean variables and expressions
- Logic gates and operators
- Truth tables
- Sum of products
- Boolean laws and identities
- Karnaugh maps
- Simplifying Boolean expressions
- D-type flip flops
- Half adders and full adders
- Logic circuits from truth tables

Term 2

- Data Protection Act
- Computer Misuse Act
- Copyright, Designs and Patents Act
- Regulation of Investigatory Powers Act
- Social impacts of computing
- Ethical considerations in computing scenarios
- Structured written responses (PEEL format)
- Algorithm design principles
- Decomposition of problems
- Flowcharts and structure diagrams
- Big O notation
- Time complexity
- Space complexity
- Constant
- Linear
- Polynomial
- Exponential
- Logarithmic

Term 3

• Stacks • Queues • Circular queues Linked lists • Binary trees • Tree traversal • Writing algorithms in pseudocode • Implementing data structures in Python • Linear search • Binary search • Preconditions for binary search • Comparing algorithm performance • Bubble sort • Insertion sort • Quick sort • Merge sort
Term 4
• Dijkstra's algorithm • A* algorithm • Shortest path problems • Heuristics in algorithms • Applying programming techniques to data structures • Python implementation of trees, graphs and nodes • Arrays • Dictionaries • Records and tuples • Object-oriented programming basics • Objects and classes • Attributes and constructors • Linked structures
Term 5
Revision & Refresh
Term 6

Independent learning

Students will be set 60 minutes of independent study every week. The nature of these activities will include the use of Smart Revise.

Students should endeavour to complete and submit the independent study tasks set by their teacher to ensure that they embed key learning, develop a good work ethic and to allow them to make as much progress as possible.

Assessment

Assessment will take place every lesson through Do Now Retrieval Practice questions, teacher questioning and exit tickets. In addition to this, students will be given ample opportunity to show their knowledge and understanding of key concepts through the completion of key, pre-identified assessment tasks. These assessment tasks will be in the form of past paper questions at the end of every half term.

At the end of the course, students will sit two exams. Each is exam is 150 minutes long and is worth 140 marks. There is also a coursework element in the form of a Project, this is worth 70 marks (20% of the overall mark)

Useful websites and reading materials