

# Curriculum Overview – COMPUTER SCIENCE



## Key Stage 3

Year 7	Year 8	Year 9
<ol style="list-style-type: none"> <li>1. School network, E-safety, Security and Ethics</li> <li>2. Computer systems</li> <li>3. Computational Thinking</li> <li>4. Programming 1 and 2 (Scratch)</li> <li>5. Internet (Basic)</li> <li>6. Modelling Data (Spreadsheets)</li> </ol>	<ol style="list-style-type: none"> <li>1. HTML Web Design CSS Java Script</li> <li>2. Programming Python</li> <li>3. Networks /Internet</li> <li>4. Computer systems: Application /system software, System architecture/ Hardware</li> <li>5. Data information and knowledge - Binary Arithmetic, Hexadecimal Data representation. Images, Sound, Compression</li> <li>6. Mobile App development</li> </ol>	<p><b>Programming</b></p> <ul style="list-style-type: none"> <li>• how to use pseudo code to create algorithms that then are converted into programs in Python.</li> <li>• concepts of programming and will build upon on the prior knowledge of the previous lessons.</li> <li>• create an entirely independent program using the knowledge obtained from the previous lessons.</li> </ul> <p><b>Internet</b></p> <ul style="list-style-type: none"> <li>• advance query searches using a search engine and know how search engines rank search results.</li> <li>• data transmission between digital computers over networks, including the internet i.e. IP addresses and packet switching.</li> <li>• the purpose of the hardware and protocols associated with networking computer systems.</li> </ul> <p><b>Media:</b></p> <ul style="list-style-type: none"> <li>• animations Creating 3D animations through object manipulation, and tweaking and adjusting lighting and camera angles.</li> </ul>



		<b>Cyber Security</b> <ul style="list-style-type: none"><li>• Identifying how users and organisations can protect themselves from cyberattacks.</li></ul>
		<b>Physical computing</b> <ul style="list-style-type: none"><li>• Sensing and controlling with the micro:bit./ Crumble kits</li></ul>

## Key Stage 4 – GCSE AQA Exam Board:



Year 10	Year 11
<p><b>Fundamentals of Algorithms</b></p> <ul style="list-style-type: none"> <li>state what is meant by an algorithm, abstraction, decomposition.</li> <li>linear and binary search, Merge and Bubble Sort</li> <li>use a flowchart or pseudocode to define the steps in a simple algorithm</li> </ul> <p><b>Programming</b></p> <ul style="list-style-type: none"> <li>identify and use variable types integer, real, Boolean, character and string</li> <li>identify variables and constants in a program</li> <li>use arithmetic operations including mod and div</li> <li>show the results of basic string manipulation functions</li> <li>use random number generation</li> <li>follow through pseudocode solutions to simple problems involving sequence, selection and iteration</li> <li>write pseudocode solutions to simple problems involving sequence, selection and iteration</li> <li>use Boolean operations NOT, AND and OR within conditions for iterative and selection structures</li> <li>read from and write to a text file</li> <li>use two-dimensional arrays in the design of solutions to simple problems</li> </ul> <p><b>Data representation</b></p> <ul style="list-style-type: none"> <li>Explain why all data needs to be converted to binary before the computer can process it</li> <li>Convert positive decimal whole numbers (0-255) into 8-bit binary numbers and vice versa</li> <li>Convert between binary and hexadecimal</li> <li>Explain the use of binary codes to represent characters</li> </ul>	<p><b>Networks</b></p> <ul style="list-style-type: none"> <li>explain the advantages of networking stand-alone computers into a local area network</li> <li>explain the difference between a LAN, a WLAN, a WAN and a PAN</li> <li>give examples of some protocols and security precautions used on networks</li> <li>describe, using diagrams or otherwise the star and bus network topologies</li> <li>describe the differences between a local area network and a wide area network such as the Internet</li> <li>describe the nature of the Internet as a worldwide collection of computer networks</li> <li>identify different transmission media</li> </ul> <p><b>Programming</b></p> <ul style="list-style-type: none"> <li>Extension of Year 10 work including GCSE project</li> </ul> <p><b>Cyber Security</b></p> <ul style="list-style-type: none"> <li>list threats to network systems including cyber attack, damage or unauthorised access</li> <li>describe briefly different forms of cyber attack such as phishing, pharming and shoulder surfing</li> <li>list ways of recognising a phishing attack</li> <li>give examples of malware and state how it can be protected against</li> <li>list security measures such as password systems, biometric methods, CAPCHA</li> <li>describe the use of biometric methods of authentication</li> <li>explain what penetration testing is and what it is used for</li> </ul> <p><b>Impacts of Digital Technology</b></p>

- Understand the term 'character set'
- Explain the relationship between the number of bits per character and the number of characters which can be represented
- Explain the representation of an image as a series of pixels represented in binary
- Explain how sound can be sampled and stored in digital form
- Perform a binary shift
- Explain the need for compression

#### **Computer Systems**

- Construct truth tables for AND, OR and NOT gates
- Construct truth tables for simple logic circuits
- Create simple logic circuit diagrams
- Define the terms hardware and software
- Explain what is meant by system software and application software
- Understand the need for, and functions of operating systems and utility programs
- Understand that the operating system handles management of the processor, memory, I/O devices, applications and security
- Explain the Von Neumann architecture
- Explain the role and operation of main memory, the ALU, control unit, clock and bus
- Explain the effects of clock speed, number of cores, cache size and cache type on CPU performance
- Understand the Fetch-Execute cycle
- Understand the difference between main memory and secondary storage
- Understand the differences between RAM and ROM
- Understand why secondary storage is required

- evaluate the impact of and issues related to the use of computers in society

- |   |  |
|---|--|
| <ul style="list-style-type: none"><li>• Understand the operation of solid state, optical and magnetic storage</li><li>• Explain the advantages and disadvantages of solid state, optical and magnetic storage</li><li>• Explain the term 'cloud storage'</li><li>• Explain the advantages and disadvantages of cloud storage when compared to local storage</li><li>• Understand the term 'embedded system' and explain how it differs from a non-embedded system</li></ul> |  |
|---|--|

## Key Stage 5 – A Level AQA Exam Board:



Year 12	Year 13
<p><b>Programming</b></p> <ul style="list-style-type: none"><li>• explain the difference between a variable and a constant</li><li>• write a pseudocode solution for a simple problem involving iteration and selection</li><li>• use nested selection and iteration statements</li><li>• use arithmetic operations and Boolean operations NOT, AND and OR</li><li>• use functions and library subroutines including random number generation</li><li>• know how to define and call a subroutine in a program</li><li>• construct algorithms using one-dimensional arrays</li><li>• read from and write to a simple text file</li><li>• write complex algorithms involving data structures, subroutines and file-handling</li><li>• interpret complex algorithms and determine the output</li></ul> <p><b>Data Representation</b></p> <ul style="list-style-type: none"><li>• differentiate between the character code for a digit and its pure binary representation</li><li>• know why Unicode was introduced</li><li>• explain how errors are detected in transmission</li><li>• subtract binary values using two's complement</li><li>• use fixed point binary to represent numbers with a fractional part</li><li>• demonstrate metadata in the properties of a data file</li><li>• apply Nyquist's theorem appropriately</li><li>• use lossy compression methods to reduce file size</li><li>• calculate the storage requirements for a bitmap image</li><li>• calculate sound sample sizes</li></ul>	<p><b>Data Structures</b></p> <ul style="list-style-type: none"><li>• describe the concept and uses of a queue, stack, list, graph, tree, hash table, dictionary and vector</li><li>• describe typical uses of these data structures</li><li>• define a rooted tree and a binary tree</li><li>• be able to apply a simple hashing algorithm</li><li>• describe what is meant by a collision and how collisions are handled using rehashing</li><li>• perform vector addition and scalar multiplication</li><li>• describe the creation and maintenance of data within queues, stacks and hash tables</li><li>• describe the characteristics of an array-based queue, circular queue and priority queue</li><li>• know how an adjacency matrix and an adjacency list may be used to represent a graph</li><li>• be able to compare the use of adjacency matrices and adjacency lists</li><li>• be able to describe and apply a number of different hashing algorithms</li><li>• describe different notations for specifying a vector</li><li>• calculate the dot product of two vectors</li><li>• describe and apply the following operations to a linear, circular and priority queue: add an item, remove an item, test for empty queue, test for full queue</li><li>• describe and apply the following operations to a stack: push, pop, peek or top, test for empty stack, test for full stack</li><li>• describe the convex combination of two vectors</li><li>• generate parity given two vectors <math>u</math> and <math>v</math> over <math>GF(2)</math></li></ul>

- know why the Vernam cipher is considered to have perfect security

### Hardware and Software

- define the terms hardware and software and explain the relationship between them
- Explain what is meant by system software and application software
- Describe some of the functions of operating systems and utility programs
- State with examples what is meant by high- and low-level languages
- Identify machine code and assembly code as low-level languages
- Explain why program translators are needed
- Explain the difference between source and object code
- Interpret simple assembly code programs
- construct truth tables for a variety of logic gates
- draw and interpret logic gate circuit diagrams involving multiple gates
- write a Boolean expression for a given logic gate circuit
- draw an equivalent logic gate circuit for a given Boolean expression
- Explain the need for, and attributes of, different types of software
- Describe the functions of operating systems, utility programs, libraries and translators
- Describe the functions of an operating system: resource management, managing hardware to allocate processors, memories and I/O devices among competing processes
- Describe the advantages and disadvantages of machine code and assembly language programming compared with high-level programming

### Algorithms

- state the essential characteristics of a recursive algorithm
- insert items into a binary search tree
- state the order in which nodes are visited in pre-order, in-order and post-order tree traversals
- give examples of linear, polynomial, exponential and logarithmic functions
- compare two algorithms in terms of efficiency
- explain the principles of a linear and binary search
- state a possible order in which nodes are visited in depth first and breadth first graph traversals
- state applications of each graph traversal
- state the purpose and applications of Dijkstra's shortest path algorithm
- Describe the Travelling Salesman problem
- Explain what is meant by a tractable or intractable problem

### Regular languages

- Interpret finite state machines with and without output
- Define a set by listing its members
- Calculate a subset, membership, union, intersection, and difference of given sets
- Form and use simple regular expressions for string manipulation and matching
- Explain the structure of a simple Turing machine.
- Read BNF production rules and validate input strings.
- Convert simple infix form to Reverse Polish Notation and vice versa

### The Internet

- Understand the structure of the Internet
- Describe the term 'Uniform Resource Locator' in the context of networking

- Explain the difference between compilation and interpretation, and describe situations when both would be appropriate
- Write simple assembly code programs
- Use de Morgan's laws to manipulate and simplify Boolean expressions
- Explain that the role of the operating system is to create a virtual machine by hiding complexities of the hardware from the user
- Explain why an intermediate language such as bytecode is produced as the final output by some compilers and how it is subsequently used
- Write assembly code programs involving arithmetic, data transfer and compare and branch instructions using a given format

#### **Computer Organisation and architecture**

- describe the role of the processor, main memory, buses and I/O controllers and how they relate to each other
- describe the Fetch-Execute cycle
- write assembly code to perform tasks involving data transfer, arithmetic operations, comparison and branch instructions
- apply immediate and direct addressing modes
- be able to suggest suitable applications for a range of input and output devices
- compare the capacity and speed of access of various media
- explain the difference between von Neumann and Harvard architectures and describe where each is typically used
- devise suitable masks and logical operations to perform given tasks in assembly language
- describe the principles of operation of a range of input, output and storage devices

- Explain the terms 'domain name' and 'IP address'
- Understand the purpose and function of the Domain Name Server (DNS) system
- Understand the role of packet switching and routers
- Consider where and why routers and gateways are used
- Understand how a firewall works
- Explain symmetric and asymmetric encryption and key exchange
- Discuss worms, Trojans and viruses and the vulnerabilities that they exploit
- Discuss how improved code quality, monitoring and protection can be used against such threats
- Describe the roles of the four layers in the TCP/IP protocol stack
- Describe the role of sockets in the TCP/IP stack
- Be familiar with MAC addresses
- Be familiar with transferring files using FTP as an anonymous and non-anonymous user
- Know that an IP address is split into a network identifier and a host identifier part
- Know that there are currently two standards of IP address, (v4 and v6) and why v6 was introduced
- Distinguish between routable and non-routable IP addresses
- Be familiar with the client server model
- Compare and contrast thin-client computing with thick-client computing

#### **Databases**

- write an entity description for each entity in a database
- define the terms attribute, primary key, composite primary key and foreign key
- produce a simple entity relationship diagram involving two or three entities



### Communication technology

- Compare synchronous and asynchronous data transmission
- Compare physical star and bus network topology in terms of advantages and disadvantages
- Explain the wireless protocols CSMA/CA and RTS/CTS
- Be able to discuss the challenges facing legislators in the digital age
- Discuss how developments in computer science and the digital technologies have dramatically altered the shape of communications and information flows in societies

- use SQL to retrieve, update, insert and delete data from a single table
- state what is meant by a client-server database
- state a problem that can arise from concurrent access on a client-server database and name a method for overcoming it
- list tasks performed by an analyst and a designer during system design

### OOP and Functional programming

- draw and interpret a class diagram
- explain what is meant by inheritance and polymorphism
- interpret and correct a simple object-oriented program
- explain why the object-oriented paradigm is used
- state the meaning of the *domain* and *co-domain* of a function
- give examples of first-class objects in a functional programming language
- write and interpret simple functions in Haskell
- evaluate simple functions involving map, filter, reduce or fold
- describe and apply list operations such as return head or tail of a list, create/test for empty list, append and prepend an item to a list
- describe what is meant by immutable data structures
- state the distinguishing features of Big Data: volume, velocity and variety
- identify nodes, edges and properties in graph schema
- write an object-oriented program
- Explain concepts of aggregation (composition and association)
- define what is meant by a first-class object in a functional programming language
- describe what is meant by functional composition and evaluate examples
- describe what is meant by a higher-order function
- write functions involving map, filter, reduce or fold

- describe features of functional programming which make it easier to write correct code and code which can be distributed to run across more than one server
- describe the features of Big Data: volume, velocity and variety
- describe the features of a fact-based model for representing data
- create and interpret simple graph schema for capturing the structure of a dataset

NEA:

- Independent project