


Year 11 Computer Science	Curriculum Intent: In year 11 the curriculum will reinforce students' understanding and apply the fundamental principles and concepts of Computer Science, including abstraction, decomposition, logic, algorithms, and data representation. Students are taught to further develop their ability to analyse problems in computational terms through practical experience of solving such problems, including designing, writing and debugging programs. Additionally, they will be encouraged to think creatively, innovatively, analytically, logically and critically and to understand the components that make up digital systems, and how they communicate with one another and with other systems. Furthermore, students are taught to understand the impacts of digital technology to the individual and to wider society and to apply mathematical skills relevant to Computer Science.			
Computer Science Year 11:	Term 1:	Term 2:	Term 3	
Topic Titles (in order of delivery)	<ul style="list-style-type: none"> • 2.1 Designing, creating and refining algorithms • 2.1 Programming Fundamentals • OCR Reference Language • 2.1 Searching and sorting algorithms • 1.5 Operating systems • 1.5.2 Utility software • 1.6 Ethical, legal, cultural and environmental impact • 1.1 Architecture of the CPU • 1.1 CPU Performance • 1.1 Systems Architecture CPU buses/memory • 1.1 Systems Architecture embedded systems • 2.3 Defensive design • 2.3 Testing 	<ul style="list-style-type: none"> • 1.1 Systems Architecture embedded systems • 1.2 Memory and Storage - Primary storage • 1.2 Secondary storage • 1.2 Units • 1.2 Data Storage • 1.4 Threats to computer systems • 1.4 Identifying and preventing vulnerabilities • 1.6 Ethical, legal, cultural and environmental impact • 1.1 Systems Architecture CPU • 1.1 Systems Architecture CPU buses/memory • 1.1 Systems Architecture embedded systems • Algorithms - OCR Reference Language/Programming Fundamentals • 2.4 Boolean Logic 	<ul style="list-style-type: none"> • 2.5 Languages • 2.1 Designing, creating and refining algorithms • 2.5.2 The Integrated Development Environment (IDE) • 2.2 Data types • 2.4 Boolean logic 	
Key knowledge / Retrieval topics	<ul style="list-style-type: none"> • Create, interpret, correct, complete, and refine algorithms using: <ul style="list-style-type: none"> o Pseudocode o Flowcharts o Reference language/high-level programming language • The use of basic string manipulation • The use of basic file handling operations: <ul style="list-style-type: none"> - Open, Read, Write, Close • The use of records to store data • The use of SQL to search for data 	<ul style="list-style-type: none"> • The purpose and characteristics of embedded systems • Examples of embedded systems • The need for primary storage • The difference between RAM and ROM • The purpose of ROM in a computer system • The purpose of RAM in a computer system • Virtual memory <p>The need for secondary storage</p> <p>Common types of storage:</p> <ul style="list-style-type: none"> • Optical 	<p>Characteristics and purpose of different levels of programming language:</p> <ul style="list-style-type: none"> • High-level languages • Low-level languages <p>The purpose of translators</p> <p>The characteristics of a compiler and an interpreter</p> <p>Create, interpret, correct, complete, and refine algorithms using:</p> <ul style="list-style-type: none"> • Pseudocode • Flowcharts 	

	<ul style="list-style-type: none"> • The use of arrays (or equivalent) when solving problems, including both one-dimensional (1D) and two-dimensional arrays (2D) • How to use sub programs (functions and procedures) to produce structured code • Random number generation • The programming task(s) must allow them to develop skills within the following areas when programming: <ul style="list-style-type: none"> • Design • Write • Test • Refine • Standard searching algorithms: <ul style="list-style-type: none"> • Binary search • Linear search • Standard sorting algorithms: <ul style="list-style-type: none"> • Bubble sort • Merge sort • Insertion sort • The purpose and functionality of operating systems: <ul style="list-style-type: none"> • User interface • Memory management and multitasking • Peripheral management and drivers • User management • File management • The purpose and functionality of utility software <ul style="list-style-type: none"> • Utility system software: <ul style="list-style-type: none"> • Encryption software • Defragmentation • Data compression <p>Standard searching algorithms:</p> <ul style="list-style-type: none"> • Binary search • Linear search 	<ul style="list-style-type: none"> • Magnetic • Solid state <p>Suitable storage devices and storage media for a given application</p> <p>The advantages and disadvantages of different storage devices and storage media relating to these characteristics:</p> <ul style="list-style-type: none"> • Capacity • Speed • Portability • Durability • Reliability <p>The units of data storage:</p> <ul style="list-style-type: none"> • Bit • Nibble (4 bits) • Byte (8 bits) • Kilobyte (1,000 bytes or 1 KB) • Megabyte (1,000 KB) • Gigabyte (1,000 MB) • Terabyte (1,000 GB) • Petabyte (1,000 TB) <p>How data needs to be converted into a binary format to be processed by a computer</p> <p>Data capacity and calculation of data capacity requirements</p> <p>How to convert positive denary whole numbers to binary numbers (up to and including 8 bits) and vice versa</p> <p>How to add two binary integers together (up to and including 8 bits) and explain overflow errors which may occur</p> <p>How to convert positive denary whole numbers into 2-digit hexadecimal numbers and vice versa</p>	<ul style="list-style-type: none"> • Reference language/high-level programming language <p>Common tools and facilities available in an Integrated Development Environment (IDE):</p> <ul style="list-style-type: none"> • Editors • Error diagnostics • Run-time environment • Translators <p>The use of data types:</p> <ul style="list-style-type: none"> • Integer • Real • Boolean • Character and string • Casting <p>Simple logic diagrams using the operators AND, OR and NOT</p> <p>Truth tables</p> <p>Combining Boolean operators using AND, OR and NOT</p> <p>Applying logical operators in truth tables to solve problems</p>
--	---	---	---

	<p>Standard sorting algorithms:</p> <ul style="list-style-type: none"> • Bubble sort • Merge sort • Insertion sort <p>Impacts of digital technology on wider society including:</p> <ul style="list-style-type: none"> • Ethical issues <ul style="list-style-type: none"> • Legal issues • Cultural issues • Environmental issues • Privacy issues <p>Legislation relevant to Computer Science:</p> <ul style="list-style-type: none"> • The Data Protection Act 2018 • Computer Misuse Act 1990 • Copyright Designs and Patents Act 1988 • Software licences (i.e. open source and proprietary) <p>The purpose of the CPU:</p> <ul style="list-style-type: none"> • The fetch-execute cycle <p>Common CPU components and their function:</p> <ul style="list-style-type: none"> • ALU (Arithmetic Logic Unit) • CU (Control Unit) • Cache • Registers <p>Von Neumann architecture:</p> <ul style="list-style-type: none"> • MAR (Memory Address Register) • MDR (Memory Data Register) • Program Counter • Accumulator <p>How common characteristics of CPUs affect their performance:</p>	<p>How to convert binary integers to their hexadecimal equivalents and vice versa</p> <p>Binary shifts</p> <p>Images</p> <p>How an image is represented as a series of pixels, represented in binary</p> <p>Metadata</p> <p>The effect of colour depth and resolution on:</p> <ul style="list-style-type: none"> • The quality of the image • The size of an image file <p>Sound</p> <p>How sound can be sampled and stored in digital form</p> <p>The effect of sample rate, duration and bit depth on:</p> <ul style="list-style-type: none"> • The playback quality • The size of a sound file <p>Forms of attack:</p> <ul style="list-style-type: none"> • Malware • Social engineering, e.g. phishing, people as the 'weak point' • Brute-force attacks • Denial of service attacks • Data interception and theft • The concept of SQL injection <p>Common prevention methods:</p> <ul style="list-style-type: none"> • Penetration testing • Anti-malware software • Firewalls • User access levels • Passwords • Encryption • Physical security 	
--	--	---	--

	<ul style="list-style-type: none"> • Clock speed • Cache size • Number of cores <p>The purpose and characteristics of embedded systems</p> <p>Examples of embedded systems</p> <p>Defensive design considerations:</p> <ul style="list-style-type: none"> • Anticipating misuse • Authentication <ul style="list-style-type: none"> • Input validation • Maintainability: • Use of sub programs • Naming conventions • Indentation • Commenting <p>The purpose of testing</p> <ul style="list-style-type: none"> • Types of testing: • Iterative • Final/terminal <p>Identify syntax and logic errors</p> <p>Selecting and using suitable test data:</p> <ul style="list-style-type: none"> • Normal • Boundary • Invalid/Erroneous <p>Refining algorithms</p>	<p>Impacts of digital technology on wider society including:</p> <ul style="list-style-type: none"> • Ethical issues • Legal issues • Cultural issues • Environmental issues • Privacy issues <p>Legislation relevant to Computer Science:</p> <ul style="list-style-type: none"> • The Data Protection Act 2018 • Computer Misuse Act 1990 • Copyright Designs and Patents Act 1988 • Software licences (i.e. open source and proprietary) <p>The purpose of the CPU:</p> <ul style="list-style-type: none"> • The fetch-execute cycle <p>Common CPU components and their function:</p> <ul style="list-style-type: none"> • ALU (Arithmetic Logic Unit) • CU (Control Unit) • Cache • Registers <p>The purpose and characteristics of embedded systems</p> <p>Examples of embedded systems</p> <p>Create, interpret, correct, complete, and refine algorithms using:</p> <ul style="list-style-type: none"> • Pseudocode • Flowcharts • Reference language/high-level programming language 	
--	---	---	--

		<p>Simple logic diagrams using the operators AND, OR and NOT</p> <p>Truth tables</p> <p>Combining Boolean operators using AND, OR and NOT</p> <p>Applying logical operators in truth tables to solve problems</p>	
Assessments	<ul style="list-style-type: none"> October CAT 	<ul style="list-style-type: none"> Comp1 PPE Comp 2 PPE 	<ul style="list-style-type: none"> In class Unit Assessments