



COMPUTER SCIENCE

TRANSITION STAGE

YEAR 7	ROTATION 1	ROTATION 2
CONTENT	Induction Emailing, internet searching, e-safety, computer components, networking, binary Game Maker Analysing games, using Game Maker/GML language to create a game	Dragons' Den Creating a game cover, understanding finances, advertising game
SKILLS	Effective use of email and search technologies, using technology responsibly, recognise acceptable/unacceptable online behaviour, understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems, understand how numbers can be represented in binary Understanding of aim and audience, use of gaming and programming concepts such as sprite, objects, events, actions, using logical reasoning to detect and correct errors, use of a textual programming languages to solve a computational problems;	Desktop publishing and design skills, numerical and spreadsheet skills, presentation and oracy skills
ASSESSMENT	Computer components, networks and binary assessment Game development practical project	Spreadsheet assessment DTP and PowerPoint assessment
USEFUL RESOURCES / GUIDANCE: https://www.thinkuknow.co.uk/ https://certificate.onlinesafetyalliance.org/ http://www.teach-ict.com/ http://www.yoyogames.com/gamemaker/windows https://www.bbc.co.uk/bitesize/subjects/zvc9q6f https://idea.org.uk/		



Curriculum & Assessment Map

FOUNDATION STAGE

YEAR 8	AUTUMN 1	AUTUMN 2	SPRING 1	SPRING 2	SUMMER 1	SUMMER 2
CONTENT	Control and Programming Understanding basic commands in programming environments such as LOGO.	Control and Programming Understanding basic commands in programming by using the Small BASIC programming environment to apply syntax	Control and Programming - Python - Understand the structural components of a program (variable declarations, command sequences, selection, iteration, data structures, subprograms) - Understand how to write code that responds appropriately to user input	Data - Data representation - Binary, HEX and ASCII Representation - Graphics and Sound representation - Data storage and compression	Graphics creation Understanding and creating different types of graphics solution to small real-world examples.	HTML and CSS Website development using HTML, CSS and Javascript
SKILLS	Understanding and applying syntax and errors	Applying syntax and sequencing, programming constructs and variables and understanding algorithms and pseudocode Understanding and applying syntax and errors, using concatenation and datatypes, using different constructs in programming, understanding sub-procedures and functions, applying all skills into a project	Understanding and applying syntax and errors. Be able to use different constructs in programming, understanding the use of variable, sub-procedures and functions, applying all skills into a project	- Understand that computers use binary to represent data (numbers, text, sound, graphics) - Be able to convert binary, hexadecimal and between the terms 'bit, nibble, byte, kilobyte (KB), megabyte (MB), gigabyte (GB), terabyte (TB)' - Understand how computers encode characters [ASCII] - Understand how bitmap images and sound are represented in binary - Understand that file storage is measured in bytes and be able to calculate file sizes - Understand how a lossless, run-length encoding algorithm works	Understanding vector and bitmap graphics, using Photoshop software and different tools to create and edit graphics	- Understand what HTML and CSS are used for - Be able to explain and implement HTML tags - Be able to define hyperlinks, explain their use and create a link using tags - Be able to describe what inline CSS is and use style attributes - Use CSS stylesheet in a high-level language - Be able to understand and implement javascript into a webpage
ASSESSMENT	LOGO programming assessment	Small BASIC programming assessment	Python programming assessment	Data representation assessment	Graphic development practical assessment	HTML and websites assessment
USEFUL RESOURCES / GUIDANCE: http://www.codecademy.com/ https://www.python.org/ http://www.bcbasic.co.uk/bcbasic.html http://www.teach-ict.com/ https://www.bbc.co.uk/bitesize/subjects/zvc9q6f https://www.w3schools.com/ https://idea.org.uk/						



Curriculum & Assessment Map

YEAR 9	AUTUMN 1	AUTUMN 2	SPRING 1	SPRING 2	SUMMER 1	SUMMER 2
CONTENT E-safety and Legal issues of IT - Understand how to use technology responsibly and recognising acceptable/unacceptable online behaviour - Understanding the legal and ethical considerations when working with technology Decomposition and abstraction - Understand how to and be able to decompose a problem - Understand how to and be able to analyse, design and program a solution to small real-world examples	Algorithms - Understanding different types of algorithms - Creating algorithms using programming constructs - Output from algorithms - Identifying errors in algorithms - Coding using algorithms - Algorithms to sort and search	Programming - Understand the structural components of a program (variable declarations, command sequences, selection, iteration, data structures, subprograms) - Understand how to write code that responds appropriately to user input - Understand the purpose and use of arithmetic, relational and logic operators	Programming - Understand how to and be able write programs in a high-level programming language - Understand the, need use and benefits of using subprograms	Computers and Software - Components of a computer - Fetch-decode-execute cycle - Primary storage	Computers and Software - Secondary storage - Operating Systems - Utility software	
SKILLS - Be aware of the risks and e-safety issues and impacts of working with technology - Be aware of ethical and legal issues arising from the use of computers - Be able to analyse a problem, investigate requirements (inputs, outputs, processing, initialisation) and design solutions - Be able to decompose a problem into smaller sub-problems - Understand how abstraction can be used effectively to model aspects of the real world - Be able to program abstractions of real-world examples	- Be able to use and interpret algorithms (flowcharts, pseudocode, written descriptions, program code) - Create algorithms to solve a particular problem, making use of programming constructs (sequence, selection, iteration) and using appropriate conventions (flowchart, pseudo-code, written description, draft program code) - Determine the correct output of an algorithm for a given set of data - Identify and correct errors in algorithms - Code an algorithm in a high-level language - Use and implement sorting and searching	- Be able to use sequencing, selection and iteration constructs in their programs - Be able to use, data types (integer, real, Boolean, char) and data structures (records, one-dimensional arrays, two-dimensional arrays) - Be able to manipulate strings - Use global/local variables and constants in programming - Be able to write code that reads/writes from/to a text file - Use arithmetic operators (add, subtract, divide, multiply, modulus, integer division) - Use relational operators (equal to, less than,	- Produce programs that are easy to read and be able to use techniques (comments, descriptive names (variables, constants, subprograms), indentation) to improve readability - Be able to differentiate between types of error in programs (logic, syntax, runtime) - Be able to design and use test plans and test data (normal, boundary, erroneous) - Be able to interpret error messages and identify, locate and fix errors in a program - Be able to determine what value a variable will hold at a given point in a program	- Identify the components of a computer with respect to: input devices, output, storage devices - Be able to explain the role of the fetch-decode-execute cycle. - Identify the hardware components used in the von Neumann architecture and explain their role in the fetch-decode-execute cycle - Be able to explain the purpose of RAM, Cache and ROM	- Be able to explain the purpose and key characteristics of secondary storage such as Magnetic storage, Optical storage and Flash Memory - Identify a range of operating systems, including Open Source and Proprietary - Be able to explain the operating systems are needed and the basic functions of an operating systems - Be able to explain the purpose of different system utilities	



Curriculum & Assessment Map

		algorithms (bubble sort, merge sort, linear search, binary search)	greater than, not equal to, less than or equal to, greater than or equal to - Use logic operators (AND, OR, NOT)	- Be able to determine the strengths and weaknesses of a program and improve it - Be able to write code that uses user-written and pre-existing subprograms, procedures, and functions - Be able to create subprograms that use parameters		
ASSESSMENT	E-Safety and ICT Laws assessment	Algorithm development assessment	Python programming assessment	Python programming assessment	Computer components (Hardware) assessment	Computer components (Software) assessment
USEFUL RESOURCES / GUIDANCE: http://www.codecademy.com/ https://www.python.org/ http://www.teach-ict.com/ https://www.bbc.co.uk/bitesize/subjects/zvc9q6f						



Curriculum & Assessment Map

EXAMINATION STAGE

YEAR 10	AUTUMN 1	AUTUMN 2	SPRING 1	SPRING 2	SUMMER 1	SUMMER 2
CONTENT	<p>Decomposition and abstraction</p> <ul style="list-style-type: none"> -Modelling aspects of the real world -Subprograms <p>Algorithms</p> <ul style="list-style-type: none"> -Flowcharts, pseudocode and program code -Constructs and input, processing and output -Variables and 1D and 2D structures -Arithmetic and logical operators -Errors in algorithms <p>Developing code</p> <ul style="list-style-type: none"> -Reading and writing programs in a high-level programming language -Converting algorithms -Programming errors <p>Programming Constructs</p> <ul style="list-style-type: none"> -Sequencing, selection, repetition <p>Data types and structures</p> <ul style="list-style-type: none"> -Primitive data types and 1D and 2D structures - Variables and constants -String Manipulation <p>Input/output</p> <ul style="list-style-type: none"> - Programs that respond to user input <p>Operators</p> <ul style="list-style-type: none"> -Arithmetic, relational and logical operators <p>Subprograms</p> <ul style="list-style-type: none"> Pre-existing and user-devised procedures and functions <p>Binary</p> <ul style="list-style-type: none"> -Binary representation of numbers 	<p>Algorithms</p> <ul style="list-style-type: none"> -Flowcharts, pseudocode and program code -Constructs and input, processing and output to solve problems -Variables and 1D and 2D dimensional data structures -Arithmetic and logical operators <p>Truth tables</p> <ul style="list-style-type: none"> -Using truth tables to solve problems <p>Developing code</p> <ul style="list-style-type: none"> -Decomposition and abstraction used to analyse, understand and solve problems -Reading and writing programs in a high-level programming language -Converting algorithms into programs <p>Programming Constructs</p> <ul style="list-style-type: none"> -Structural components of programs -Sequencing, selection, repetition <p>Data types and structures</p> <ul style="list-style-type: none"> -Primitive data types and 1D and 2D-dimensional structured data types <p>Operators</p> <ul style="list-style-type: none"> Arithmetic, relational and logical operators <p>Binary</p> <ul style="list-style-type: none"> -Representation of text, sound and graphics in binary -Binary arithmetic -Hexadecimal 	<p>Decomposition and abstraction</p> <ul style="list-style-type: none"> - Using decomposition and abstraction to model aspects of the real world -Subprograms <p>Algorithms</p> <ul style="list-style-type: none"> Flowcharts, pseudocode and program code -Constructs and input, processing and output -Variables and 1D and 2D data structures <p>Developing code</p> <ul style="list-style-type: none"> -Decomposition and abstraction used to analyse, understand and solve problems -Reading and writing programs in a high-level programming language <p>Data types and structures</p> <ul style="list-style-type: none"> -Primitive data types and 1D and 2D structured data types <p>Subprograms</p> <ul style="list-style-type: none"> -Pre-existing and user-devised procedures and functions -Functions and parameters <p>Hardware</p> <ul style="list-style-type: none"> -CPU and fetch-decode-execute cycle -RAM and ROM -Secondary storage 	<p>Decomposition and abstraction</p> <ul style="list-style-type: none"> - Using decomposition and abstraction to model aspects of the real world -Subprograms <p>Algorithms</p> <ul style="list-style-type: none"> -Flowcharts, pseudocode and program code -Constructs and input, processing and output -Variables and 1D and 2D data structures -Algorithm outputs using trace tables -Sorting and Searching algorithms <p>Programming Constructs</p> <ul style="list-style-type: none"> -Sequencing, selection, repetition -Subprograms <p>Data types and structures</p> <ul style="list-style-type: none"> -Primitive data types and 1D and 2D structured data types -String manipulation <p>Input/output in programming</p> <ul style="list-style-type: none"> -Programs that accept and respond to user input -Validation in programming <p>Subprograms</p> <ul style="list-style-type: none"> -Pre-existing and user-devised procedures and functions Functions and parameters <p>Software</p> <ul style="list-style-type: none"> -Operating systems and their functions (file management, process management, peripheral 	<p>Algorithms</p> <ul style="list-style-type: none"> -Flowcharts, pseudocode and program code -Constructs and input, processing and output -Variables and 1D and 2D data structures -Sorting and Searching algorithms <p>Programming Constructs</p> <ul style="list-style-type: none"> -Sequencing, selection, repetition <p>Data types and structures</p> <ul style="list-style-type: none"> -String manipulation <p>Input/output in programming</p> <ul style="list-style-type: none"> -Programs that accept and respond to user input -Reading and writing to csv and text files -Validation in programming -Authentication in programming <p>Software</p> <ul style="list-style-type: none"> -Utility software and their functionality -Robust software and identifying vulnerabilities <p>Network security</p> <ul style="list-style-type: none"> -Network security -Identifying network vulnerabilities and protecting networks -Threats to digital systems -Malware, hackers and cyberattacks -Protecting of digital systems and data 	<p>Develop code</p> <ul style="list-style-type: none"> -Programming errors <p>Programming Constructs</p> <ul style="list-style-type: none"> -Sequencing, selection, repetition -Subprograms <p>Data types and structures</p> <ul style="list-style-type: none"> --Primitive data types and 1D and 2D structured data types -Variables and constants <p>Input/output in programming</p> <ul style="list-style-type: none"> -Programs that accept and respond to user input <p>Operators</p> <ul style="list-style-type: none"> -Arithmetic, relational and logical operators <p>Subprograms</p> <ul style="list-style-type: none"> -Pre-existing and user-devised procedures and functions -Functions and parameters <p>Networks</p> <ul style="list-style-type: none"> -Different types of networks (LAN, WAN) -Structure of the internet -Wired and wireless connectivity -Network speeds -Network protocols -Network topologies



Curriculum & Assessment Map

	<ul style="list-style-type: none"> -Binary arithmetic Data representation, storage and compression -Limitations of binary representation -Data storage, file calculation and data capacity requirements 	<ul style="list-style-type: none"> Data representation, storage and compression -ASCII -Image and sound data representation -Limitations of binary representation -Data storage, file calculation and data capacity requirements -Data compression 		<ul style="list-style-type: none"> management, user management) -Utility software and their functionality 		
SKILLS	<ul style="list-style-type: none"> -Use decomposition and abstraction to model aspects of the real world -Follow and write algorithms (flowcharts, pseudocode*, program code) that use sequence, selection, repetition and iteration and input, processing and output -Follow and write algorithms that use variables and constants and 1D and 2D structures -Follow and write algorithms that use arithmetic, relational and logical operators -Identify the types of errors that can occur in programs and be able to identify and correct logic errors in algorithms -Use decomposition and abstraction to analyse, understand and solve problems -Read, write, analyse and refine programs written in a high-level programming language -Convert algorithms into programs 	<ul style="list-style-type: none"> -Follow and write algorithms (flowcharts, pseudocode*, program code) that use sequence, selection, repetition and iteration and input, processing and output -Follow and write algorithms that use variables and constants and 1D and 2D dimensional data structures -Follow and write algorithms that use arithmetic, relational and logical operators -Apply logical operators in truth tables with up to three inputs -Use decomposition and abstraction to analyse, understand and solve problems -Read, write, analyse and refine programs written in a high-level programming language -Convert algorithms into programs -Identify the structural components of programs -Write programs that make appropriate use of 	<ul style="list-style-type: none"> -Use decomposition and abstraction to model aspects of the real world and analyse, understand and solve problems -Understand the benefits of using subprograms -Follow and write algorithms (flowcharts, pseudocode*, program code) that use sequence, selection, repetition and iteration and input, processing and output to solve problems -Follow and write algorithms that use variables and constants and 1D and 2D data structures -Read, write, analyse and refine programs written in a high-level programming language -Write programs that make appropriate use of primitive data types and 1D and 2D structured data types -Write programs that use pre-existing and user-devised subprograms (procedures, functions) -Understand the von Neumann stored program concept and the role of 	<ul style="list-style-type: none"> -Use decomposition and abstraction to model aspects of the real world and analyse, understand and solve problems -Understand the benefits of using subprograms -Follow and write algorithms (flowcharts, pseudocode*, program code) that use sequence, selection, repetition and iteration and input, processing and output to solve problems -Follow and write algorithms that use variables and constants and 1D and 2D data structures -Determine the correct output of an algorithm using a trace table -Be able to explain and use different search and sort algorithms (bubble sort, merge sort, linear search, binary search) -Write programs that make appropriate use of sequencing, selection, repetition, iteration and single entry/exit points from code blocks and subprograms 	<ul style="list-style-type: none"> -Follow and write algorithms (flowcharts, pseudocode*, program code) that use sequence, selection, repetition and iteration and input, processing and output to solve problems -Follow and write algorithms that use variables and constants and 1D and 2D dimensional data structures --Be able to explain and use different search and sort algorithms (bubble sort, merge sort, linear search, binary search) -Write programs that make appropriate use of sequencing, selection, repetition (count-controlled, condition-controlled), iteration (over every item in a data structure) and single entry/exit points from code blocks and subprograms -Write programs that manipulate strings (length, position, substrings, case conversion) 	<ul style="list-style-type: none"> -Identify, locate and correct program errors (logic, syntax, runtime) -Write programs that make appropriate use of sequencing, selection, repetition (count-controlled, condition-controlled), iteration (over every item in a data structure) and single entry/exit points from code blocks and subprograms -Write programs that make appropriate use of primitive data types (integer, real, Boolean, char) 1D and 2D structured data types (string, array, record) -Write programs that make appropriate use of variables and constant -Write programs that accept and respond appropriately to user input -Write programs that use arithmetic, relational and logical operators -Write programs that use pre-existing (built-in, library) and user-devised



Curriculum & Assessment Map

<ul style="list-style-type: none"> -Use techniques (layout, indentation, comments, meaningful identifiers, white space) to make programs easier to read, understand and maintain -Identify, locate and correct program errors -Identify the function and structural components of programs -Write programs that make appropriate use of sequencing, selection, repetition iteration and single entry/exit points from code blocks and subprograms -Write programs that make appropriate use of primitive data types and 1D and 2D data types -Write programs that make appropriate use of variables and constants -Write programs that manipulate strings -Write programs that accept and respond appropriately to user input -Write programs that use arithmetic operators -Be able to write programs that use pre-existing and user-devised subprograms (procedures, functions) -Understand that computers use binary to represent data and program instructions and be able to determine the maximum number of states that can be 	<ul style="list-style-type: none"> sequencing, selection, repetition iteration and single entry/exit points from code blocks and subprograms -Write programs that make appropriate use of primitive data types and 1D and 2D structured data types -Write programs that use relational and logical operators - Understand that computers use binary to represent data (numbers, text, sound, graphics) and program instructions and be able to determine the maximum number of states that can be represented by a binary pattern of a given length -Be able to add together two positive binary patterns and apply logical and arithmetic binary shifts -Understand why hexadecimal notation is used and be able to convert between hexadecimal and binary -Understand how computers encode characters using 7-bit ASCII -Understand how bitmap images are represented in binary -Understand how analogue sound is represented in binary -Understand the limitations of binary representation of data 	<ul style="list-style-type: none"> main memory (RAM), CPU clock, address bus, data bus, control bus in the fetch-decode-execute cycle -Understand the role of secondary storage and the ways in which data is stored on devices (magnetic, optical, solid state) -Be able to construct expressions to calculate file sizes and data capacity requirements 	<ul style="list-style-type: none"> -Write programs that make appropriate use of primitive data types and 1D and 2D structured data types -Write programs that manipulate strings (length, position, substrings, case conversion) -Write programs that accept and respond appropriately to user input -Write programs that implement validation -Write programs that use pre-existing and user-devised subprograms (procedures, functions) -Understand the purpose and functionality of an operating system (file management, process management, peripheral management, user management) -Understand the purpose and functionality of utility software (file repair, backup, data compression, disk defragmentation, anti-malware) 	<ul style="list-style-type: none"> -Write programs that read from and write to comma separated value text files -Write programs that implement validation (length check, presence check, range check, pattern check) -Write programs that implement authentication (ID and password, lookup) -Understand the purpose and functionality of utility software (file repair, backup, data compression, disk defragmentation, anti-malware) -Understand the importance of developing robust software and methods of identifying vulnerabilities (audit trails, code reviews) -Understand the importance of network security, ways of identifying network vulnerabilities (penetration testing, ethical hacking) and methods of protecting networks (access control, physical security, firewalls) -Understand the threat to digital systems posed by malware (viruses, worms, Trojans, ransomware, key loggers) and how hackers exploit technical vulnerabilities (unpatched software, out-of-date anti-malware) and use social engineering to carry out cyberattacks -Understand methods of protecting digital systems 	<ul style="list-style-type: none"> subprograms (procedures, functions) -Write programs that make appropriate use of global and local variables -Understand why computers are connected in a network -Understand different types of networks (LAN, WAN) -Understand how the internet is structured (IP addressing, routers) -Understand how the characteristics of wired and wireless connectivity impact on performance (speed, range, latency, bandwidth) -Understand that network speeds are measured in bits per second (kilobit, megabit, gigabit) and be able to construct expressions involving file size, transmission rate and time -Understand the role of and need for network protocols (Ethernet, Wi-Fi, TCP/IP, HTTP, HTTPS, FTP) and email protocols (POP3, SMTP, IMAP) -Understand characteristics of network topologies (bus, star, mesh) -Be able to construct expressions to calculate file sizes and data capacity requirements
---	--	--	---	--	--



Curriculum & Assessment Map

	<p>represented by a binary pattern of a given length</p> <ul style="list-style-type: none"> -Understand how computers represent and manipulate unsigned integers and two's complement signed integers -Be able to convert between denary and 8-bit binary numbers -Be able to add together two positive binary patterns and apply logical and arithmetic binary shifts -Understand the concept of overflow in relation to the number of bits available to store a value -Understand the limitations of binary representation of data when constrained by the number of available bits -Understand that data storage is measured in binary multiples and be able to construct expressions to calculate file sizes and data capacity requirements 	<p>when constrained by the number of available bits</p> <ul style="list-style-type: none"> -Understand that data storage is measured in binary multiples (bit, nibble, byte, kibibyte, mebibyte, gibibyte, tebibyte) and be able to construct expressions to calculate file sizes and data capacity requirements -Understand the need for data compression and methods of compressing data) 			<p>and data (anti-malware, encryption, acceptable use policies, backup and recovery procedures)</p>	
ASSESSMENT	<p>In class practice paper questions on topics of:</p> <ul style="list-style-type: none"> -Decomposition and Algorithms -Binary Representation 	<p>In class practice paper questions on topics of:</p> <ul style="list-style-type: none"> -Algorithms and Programming -Data Representation 	<p>In class practice paper questions on topics of:</p> <ul style="list-style-type: none"> -Algorithms and Programming -Computer Hardware 	<p>In class practice paper questions on topics of:</p> <ul style="list-style-type: none"> -Algorithms and Programming -Computer Software 	<p>In class practice paper questions on topics of:</p> <ul style="list-style-type: none"> -Algorithms and Programming -Computer Software 	<p>In class practice paper questions on topics of:</p> <ul style="list-style-type: none"> -Algorithms and Programming -Networks and Network security
<p>USEFUL RESOURCES / GUIDANCE:</p> <p>https://qualifications.pearson.com/content/dam/pdf/GCSE/Computer%20Science/2020/specification-and-sample-assessments/GCSE_L1_L2_Computer_Science_2020_Specification.pdf http://www.codecademy.com/ https://www.python.org/ http://www.teach-ict.com/ https://www.youtube.com/channel/UC0HzEBLIJxlrwBAHJ5S9JQg/playlists?view=50&sort=dd&shelf_id=21 https://computerscienceuk.com/gcse-9-1/gcse-videos/ Edexcel GCSE (9-1) Computer Science Student Book by Ann Weidmann, David Waller, Cynthia Selby (ISBN: 9781292359991)</p>						



Curriculum & Assessment Map

YEAR 11	AUTUMN 1	AUTUMN 2	SPRING 1	SPRING 2	SUMMER 1	SUMMER 2	
CONTENT	<p>Networks</p> <ul style="list-style-type: none"> - Networking - Types of networks - Connectivity and speed - Topologies - Transmission and protocols - Network Security <p>Revisiting topic - Data</p> <ul style="list-style-type: none"> - Binary - Data representation - Data storage and compression - Encryption 	<p>The internet and the world wide web</p> <ul style="list-style-type: none"> - Internet structure - WWW and its components - HTML and CSS - Server side processing <p>Revisiting topic - Data and Computers</p> <ul style="list-style-type: none"> - Databases - Machines and computational models 	<p>Revisiting topic - Algorithms</p> <ul style="list-style-type: none"> - Understanding different types of algorithms - Creating algorithms using programming constructs - Output from algorithms - Identifying errors in algorithms - Coding using algorithms - Algorithms to sort and search <p>Revisiting topic - Computers and Software</p> <ul style="list-style-type: none"> - Computer Hardware - Logic - Systems Software - Applications Software - Programming languages 	<p>Revisiting topic - Programming</p> <ul style="list-style-type: none"> - Understand the structural components of a program (variable and type declarations, command sequences, selection, iteration, data structures, subprograms) - Understand how to and be able write programs in a high-level programming language - Understand the, need use and benefits of using subprograms <p>The Bigger Picture</p> <ul style="list-style-type: none"> - Emerging trends, issues and impact 	<p>Revision</p>		
SKILLS	<ul style="list-style-type: none"> - Understand why computers are connected in a network and the different types of networks (LAN, WAN) and usage models (client server, peer-to-peer) - Understand wired and wireless connectivity - Understand that network data speeds are measured in bits per second [Mbps, Gbps] - understand the role of and need for network protocols - Understand that data can be transmitted over networks [TCP/IP] - Understand characteristics of network topologies - Understand the importance of network 	<ul style="list-style-type: none"> - Understand what is meant by the internet and how the internet is structured - Understand what is meant by the world wide web (WWW) and components of the WWW [web server URLs, ISP, HTTP, HTTPS, HTML] - Be able to use HTML and CSS to construct web pages [formatting, links, images, media, layout, styles, lists] - Understand the client-server model, the difference between client-side and server-side processing and the role of cookies - Understand the characteristics of 	<ul style="list-style-type: none"> - Understand how to create an algorithm to solve a particular problem, making use of programming constructs (sequence, selection, iteration) and using appropriate conventions (flowchart, pseudo-code, written description, draft program code) - Understand how to code an algorithm in a high-level language - Understand the input process-output model - Understand the function of hardware components of a computer system [processor (CPU), memory, secondary storage, input devices, output devices] and how they work together - 	<ul style="list-style-type: none"> - Be able to use sequencing, selection and iteration constructs in their programs - Understand the need for, and understand how to use, data structures (records, one-dimensional arrays, two-dimensional arrays) - Be able to write code that reads/writes from/to a text file - Understand the benefit of producing programs that are easy to read and be able to use techniques (comments, descriptive names (variables, constants, subprograms), indentation) to improve readability and to explain how the code works 			



Curriculum & Assessment Map

	<p>security and methods of identifying vulnerabilities</p> <ul style="list-style-type: none"> - Understand that computers use binary to represent data (numbers, text, sound, graphics) and program instructions - Understand how computers represent and manipulate numbers [unsigned integers, signed integers (sign and magnitude, Two's complement)] - Be able to convert between binary and denary whole numbers (0-255) and vice versa - Understand why hexadecimal notation is used and be able to convert between hexadecimal and binary and vice versa - Understand and be able to convert between the terms 'bit, nibble, byte, kilobyte (KB), megabyte (MB), gigabyte (GB), terabyte (TB)' - Understand that file storage is measured in bytes and be able to calculate file sizes - Understand how a lossless, run-length encoding algorithm works - Understand and be able to calculate the time required to transmit a file and storage requirements for files - Understand the need for data encryption - Understand how a Caesar cipher algorithm works 	<p>structured and unstructured data</p> <p>Understand that data can be decomposed and organised in a structured database [tables, records, fields, relationships, keys]</p> <ul style="list-style-type: none"> - Understand the input process-output model - Understand the function of hardware components of a computer system [processor (CPU), memory, secondary storage, input devices, output devices] and how they work together - Understand the concept of a stored program and the role of components of the processor [control unit (CU), arithmetic/ logic unit (ALU), registers, clock, address bus, data bus] in the fetch-decode-execute cycle - Understand the function of assembly code and be able to interpret assembly code 	<p>understand the function of different types of main memory (RAM, ROM, cache)</p> <ul style="list-style-type: none"> - Understand how data is stored on physical devices (magnetic, optical, solid state) - Understand the concept of storing data in the 'cloud' and other contemporary secondary storage - Understand the need for embedded systems and their functions - Be able to construct truth tables and logic statements [AND, OR, NOT] - Know what an operating system its functions - Understand the purpose and functions of utility software (managing, repairing and converting files; compression; defragmentation; backing up; anti-virus, anti-spyware) - Understand how software can be used to simulate and model aspects of the real world and be able to create software models - Understand what is meant by high-level and low-level programming languages - Understand what is meant by an assembler, a compiler and an interpreter when translating programming languages 	<ul style="list-style-type: none"> - Be able to interpret error messages and identify, locate and fix errors in a program - Understand the benefits of using subprograms - Be aware of current and emerging trends in computing technology [quantum computing, DNA computing, artificial intelligence (AI), nano technology] - Be aware of the impact of computing on individuals, society and the environment - Be aware of ethical and legal issues arising from the use of computers - Be aware of ownership issues relating to computing [intellectual property, patents, licensing, open source and proprietary software] 		
--	---	---	---	--	--	--



Curriculum & Assessment Map

ASSESSMENT	In class practice paper questions on topics of: -Networks -Data representation	In class practice paper questions on topics of: -Internet and WWW -Databases Mock exam	In class practice paper questions on topics of: -Algorithms -Computer Hardware and Software	In class practice paper questions on topics of: - Programming -Legal and Ethical issues	Walking talking practice exams	
USEFUL RESOURCES / GUIDANCE: https://qualifications.pearson.com/en/qualifications/edexcel-gcse/computer-science-2016.html http://www.codecademy.com/ https://www.python.org/ http://www.teach-ict.com/ https://www.youtube.com/channel/UC0HzEBLIJxlwBAHJ5S9JQg https://computerscienceuk.com/gcse-9-1/gcse-videos/ Edexcel GCSE (9-1) Computer Science Student Book by Ann Weidmann, David Waller, Alex Hadwen-Bennett, Chris Charles (ISBN: 978-1292125886)						



Curriculum & Assessment Map

ADVANCED STAGE: COMPUTER SCIENCE A LEVEL

YEAR 12	AUTUMN 1	AUTUMN 2	SPRING 1	SPRING 2	SUMMER 1	SUMMER 2
CONTENT	Component 1.1: The characteristics of contemporary processors Input, output and storage devices Component 1.2.3: Introduction to programming Component 2.2.1: Programming techniques	Component 1.2.1: Operating Systems Component 1.3.1: Databases Component 1.2.3: Assembly language Component 1.4.2: Data structures Component 2.3: Algorithms	Component 1.4.1: Data Types Component 1.5.1: Computing related legislation Component 1.2.3: Assembly language Component 2.3: Algorithms Component 2.1: Elements of computational thinking	Component 1.5.2: Ethical, moral and cultural issues Component 1.3.2: Networks Component 1.2.2: Applications generation Component 1.3.3: Web Technologies Component 2.1: Elements of computational thinking Component 1.3.3: Boolean Algebra	Component 1.2.2: Applications generation Component 1.3.3: Boolean Algebra Component 2.2.2: Software Development	Component 3.1: Analysis of the problem Component 1.2.4: Types of Programming Language. Component 3: Introduction to Monkey-x
SKILLS	Develop deeper knowledge and understanding of: -Structure and function of the processor -Types of processor -Input, output and storage Practical work: -Write code in Python and Visual Basic to solve problems. Develop knowledge on programming structure such as: -Programming constructs -Global and local variables Modularity -Functions and procedures -Parameter passing by value and reference -Use of an IDE to develop/debug a program	Develop deeper knowledge and understanding of: -The need for, function and purpose of operating systems. -Memory Management -Interrupts Service Routines (ISR) and its role within the fetch decode execute cycle. -Scheduling - BIOS Develop knowledge and understanding of: -Relational database -flat file and how to capture, manage and exchange data. Practical Work: -Write code in Assembly language -Write code in python using data structures to solve problem. -Write algorithms to solve simple problems.	Develop knowledge and understanding and carry out activities on: - Represent positive integers in binary. -Use of sign and magnitude and two's complement to represent negative numbers in binary. -Addition and subtraction of binary integers. -Represent positive integers in hexadecimal -Convert Hexadecimal and binary and denary. -Positive and negative real numbers using normalised floating point representation. -How character sets (ASCII and UNICODE) are used to represent text. -data protection, data misuse, copyright and investigatory powers. -searching and sorting algorithms. - standard algorithms for add, delete items from stack and queue	Develop knowledge and understanding of: -Legal, moral, ethical and cultural issues. - Characteristics of networks and the importance of protocols and standards. -Internet structure: The TCP/IP Stack. DNS -Protocol layering. -LANs and WANs. -Packet and circuit switching. -Client-server and Peer to peer. Develop knowledge and understanding of: -The nature of applications, justifying suitable applications for a specific purpose. -Utilities. Practical work to develop website using: - html, CSS and Java Script. Develop knowledge and understanding of Lossy v lossless compression.	Develop knowledge and understanding of: -Waterfall lifecycle, agile methodologies, extreme programming, the spiral model and rapid application development. - The relative merits and drawbacks of different methodologies and when they might be used. - Writing and following algorithms. - Different test strategies, including black and white box testing and alpha and beta testing. -Test programs that solve problems using suitable test data and end user feedback, justify a test strategy for a given situation. Develop knowledge and understanding of: -Open source vs Closed source. -Translators: interpreters, compilers and assemblers.	-Investigate the features available on sample games and record the features it offers. -Research and find a suitable problem(game) to make a start on the project. -Define the stakeholder for the problem. Develop knowledge and understanding of: -Need for and characteristics of a variety of programming paradigms. -Procedural languages. -Object-oriented languages Practical work on Monkey-x programming language to develop games.



Curriculum & Assessment Map

			-suitability of algorithms for solving a problem. Develop knowledge and understanding of: Thinking abstractly and Thinking ahead		-Boolean expressions, Karnaugh maps.	
ASSESSMENT	Weekly tests based on topics taught	November internal exam Weekly tests based on topics taught	February internal exam Weekly tests based on topics taught	Weekly tests based on topics taught	End of year internal exam	Computing Project (Component 3) - practical development

USEFUL RESOURCES / GUIDANCE:

- <http://www.ocr.org.uk/qualifications/as-a-level-gce/as-a-level-gce-computer-science-h046-h446-from-2015/>
- <http://www.codecademy.com/>
- <https://www.python.org/>
- <http://www.teach-ict.com/>
- <https://student.craigndave.org/a-level-videos>
- <https://isaacomputerscience.org/home>
- <https://www.youtube.com/playlist?list=PL8dPuuaLjXtNIUrzyH5r6jN9ullgZBpdo>

Books

OCR A Level Computer Science by Sean O'Byrne, George Rouse, Jason Pitt, published by Hodder Education (ISBN: 9781471839764)
OCR AS and A Level Computer Science by PM Heathcote and RSU Heathcote (ISBN : 979 1 910523 05 6)

YEAR 13	AUTUMN 1	AUTUMN 2	SPRING 1	SPRING 2	SUMMER 1	SUMMER 2	
CONTENT	Component 3.1: Analysis of the problem Component 1.4.1: Data Types Component 1.3.2: Databases Component 1.2.4: Types of Programming Language. Component 1.2.2: Applications Generation Component 1.3.3: Boolean Algebra Component 3.3.1: Iterative development process	Component 3.1: Analysis of the problem Component 3.2.1, 3.2.2: Decompose the problem, Describe the solution Component 2.3.1: Algorithms Component 1.4.2: Data Structures Component 3.3.1: Iterative development process Component 3.3.2: Testing to inform development	Component 2.3.1: Algorithms Component 3.2.2, 3.2.3, 3.3.2, 3.4.2, 3.4.4: Project development Describe the solution, Describe the approach to testing, Testing to inform development, Success of the solution, Maintenance and development Component 1.4.2: Data Structures	Component 2.3.1: Algorithms Component 2.1.2: Thinking ahead Component 1.4.2: Data Structures Component 1.3.4: Web Technologies Component 1.3.2: Networks	Component 2.1.5: Thinking concurrently Component 2.2.1: Programming techniques Component 2.2.2: Computational methods Component 1.3.1: Compression, Encryption and Hashing		
SKILLS	Develop knowledge and understanding of: -Representation and normalisation of floating point numbers in binary. -	Computing Project-Analysis - Decompose the game development problem, then explain and justify	Develop knowledge and understanding of: -The nature, benefits and drawbacks of caching.	Develop knowledge and understanding of: -Measures and methods to determine the	Develop knowledge and understanding of: - Determine the parts of a problem that can be tackled at the same time.		



Curriculum & Assessment Map

	<p>Floating point arithmetic, positive and negative numbers, addition and subtraction. -Bitwise manipulation and masks: shifts, combining with AND, OR, and XOR. -How character sets (ASCII and UNICODE) are used to represent text. -Normalisation of databases to 3NF -Reading, writing and interpreting SQL -Referential integrity -Transaction processing -ACID</p> <p>Analysis -Define the stakeholder for the problem. -Define the user requirement and stake holders.</p> <p>Develop knowledge and understanding of: -Procedural languages. -Assembly language -Modes of addressing memory -Stages of compilation -Linkers, Loaders and the use of libraries -The use of pipelining in a processor to improve efficiency GPUs and their uses - Use of rules to derive or simplify statements in Boolean algebra: De Morgan's Laws, distribution, association, commutation, double negation.</p>	<p>the structure of the solution. -Describe the parts of the solution using algorithms justifying how these algorithms form a complete solution to the problem.</p> <p>Develop knowledge and understanding of: -The suitability of different algorithms for a given task and data set, in terms of execution time and space. - Structures to store data: linked-list, graph (directed and undirected).</p> <p>Practical work Implementing code in Monkey-x to develop Computing project (game) -Obtain feedback from the stakeholders -Analysis of suggestions and make improvement on the coding. -Create test plan and record evidence of tests.</p>	<p>-The suitability of different algorithms for a given task and data set, in terms of execution time and space. -Structures to store data: stack, queue, tree, binary search tree, hash table.</p> <p>Complete the following for the game project: -Describe usability features to be included in the Solution. -Identify key variables / data structures / classes justifying choices and any necessary validation. Identify the test data to be used during the iterative development and post development phases and justify the choice of this test data. - Use the test evidence from the development and post development process to evaluate the solution against the success criteria from the analysis. -Provide annotated evidence of the usability features from the design, commenting on their effectiveness. -Discuss the maintainability of the solution. -Discuss potential further development of the solution.</p>	<p>efficiency of different algorithms -Big O notation -Comparison of the complexity of algorithms. -Standard algorithms (merge sort, quick sort, Dijkstra's shortest path algorithm, A* algorithm) -The nature, benefits and drawbacks of caching. -How to create, traverse, add data to and remove data from the data structures mentioned above. -Search engine indexing. PageRank algorithm. -Server and client side processing. -Network security and threats, use of firewalls, proxies and encryption. -Network hardware.</p>	<p>- Outline the benefits and trade offs that might result from concurrent processing in a particular situation. -Recursion, how it can be used and compares to an iterative approach. - Features that make a problem solvable by computational methods. -Problem recognition. -Problem decomposition. -Use of divide and conquer. -Use of abstraction. -Apply knowledge of: • backtracking • data mining • heuristics • performance modelling • pipelining • visualisation to solve problems.</p> <p>Develop knowledge and understanding of: -Object oriented techniques. -Run length encoding and dictionary coding for lossless compression -Symmetric and asymmetric encryption. -Different uses of hashing.</p>	
--	--	--	---	---	--	--



Curriculum & Assessment Map

	- The logic associated with D type flip flops, half and full adders. Practical work Implementing code in Monkey-x to develop Computing project (game). Progress recorded in report.					
ASSESSMENT	September internal exam Computing Project (Component 3) – Analysis	November internal exam Computing Project (Component 3) – Implementation	February internal exam Computing Project (Component 3) – Design and Testing	Submission of Computing coursework (Component 3 – 20%) Mock Exams	Mock exams	
<p>USEFUL RESOURCES / GUIDANCE: http://www.ocr.org.uk/qualifications/as-a-level-gce/as-a-level-gce-computer-science-h046-h446-from-2015/ http://www.codecademy.com/ https://www.python.org/ http://www.teach-ict.com/ https://student.craigndave.org/a-level-videos https://isaacomputerscience.org/home https://www.youtube.com/playlist?list=PL8dPuuaLjXtNIUrzyH5r6jN9ullgZBpdo</p> <p>Books OCR A Level Computer Science by Sean O'Byrne, George Rouse, Jason Pitt, published by Hodder Education (ISBN: 9781471839764) OCR AS and A Level Computer Science by PM Heathcote and RSU Heathcote (ISBN : 979 1 910523 05 6)</p>						



ADVANCED STAGE: BTEC NATIONALS IN INFORMATION TECHNOLOGY

YEAR 12 SINGLE	AUTUMN 1	AUTUMN 2	SPRING 1	SPRING 2	SUMMER 1	SUMMER 2
CONTENT	Unit 1: Information Technology Systems	Unit 1: Information Technology Systems	Unit 2: Creating Systems to Manage Information	Unit 2: Creating Systems to Manage Information	Unit 2: Creating Systems to Manage Information	Unit 3: Using Social Media in Business
SKILLS	Demonstrate knowledge and understanding of information technology terms, standards, concepts and processes. Select and use information technologies and procedures to explore likely outcomes and find solutions to problems in context	Demonstrate knowledge and understanding of information technology terms, standards, concepts and processes. Select and use information technologies and procedures to explore likely outcomes and find solutions to problems in context	Demonstrate knowledge of database development terminology, standards, concepts and processes. Be able to develop a database solution to meet a client brief with appropriate justification.	Demonstrate knowledge of database development terminology, standards, concepts and processes. Be able to develop a database solution to meet a client brief with appropriate justification.	Demonstrate knowledge of database development terminology, standards, concepts and processes. Be able to develop a database solution to meet a client brief with appropriate justification.	Explore the impact of social media on the ways in which businesses promote their products and services. Develop a plan and implement the use social media in a business to meet requirements
ASSESSMENT	Written and verbal feedback provided per criteria for each unit. Unit 1 and 2 are assessed externally. Unit 1 is an exam and Unit 2 takes the form of a controlled coursework which is carried out under exam conditions. The results of Unit 1 and 2 are returned as N (Near Pass), Pass, Merit and Distinction. Unit 3 is internally assessed work graded Distinction (A) , Merit(B) Pass (E). Feedback provided guides students to areas of the specification which they have not met.					
USEFUL RESOURCES / GUIDANCE: BTEC Nationals Information Technology Student Book + Active book: For the 2016 specifications (BTEC Nationals IT 2016) – ISBN – 978-1292140414 https://qualifications.pearson.com/en/qualifications/btec-nationals/computing-2016.html http://www.teach-ict.com/ https://www.knowitallninja.com/						

YEAR 12 TRIPLE	AUTUMN 1	AUTUMN 2	SPRING 1	SPRING 2	SUMMER 1	SUMMER 2
CONTENT	Unit 1: Information Technology Systems Unit 2: Creating Systems to Manage Information Unit 5: Data Modelling	Unit 1: Information Technology Systems Unit 2: Creating Systems to Manage Information Unit 5: Data Modelling	Unit 3: Using Social Media in Business Unit 4: Programming Unit 5: Data Modelling	Unit 3: Using Social Media in Business Unit 4: Programming Unit 9: IT Project Management	Unit 3: Using Social Media in Business Unit 4: Programming Unit 9: IT Project Management	Unit 6: Website Development Unit 6: Website Development Unit 9: IT Project Management
SKILLS	Demonstrate knowledge and understanding of information technology terms, standards, concepts and processes. Select and use information technologies	Demonstrate knowledge and understanding of information technology terms, standards, concepts and processes. Select and use information technologies	Explore the impact of social media on the ways in which businesses promote their products and services. Develop a plan and implement the use social media in a	Explore the impact of social media on the ways in which businesses promote their products and services. Develop a plan and implement the use social media in a	Explore the impact of social media on the ways in which businesses promote their products and services. Develop a plan and implement the use social media in a	Understand the principles of website development, design and develop a website to meet client requirements.



Curriculum & Assessment Map

	<p>and procedures to explore likely outcomes and find solutions to problems in context</p> <p>Demonstrate knowledge of database development terminology, standards, concepts and processes. Be able to develop a database solution to meet a client brief with appropriate justification.</p> <p>Investigate data modelling and how it can be used in the decision-making process, design and develop a data model to meet client requirements</p>	<p>and procedures to explore likely outcomes and find solutions to problems in context</p> <p>Demonstrate knowledge of database development terminology, standards, concepts and processes. Be able to develop a database solution to meet a client brief with appropriate justification.</p> <p>Investigate data modelling and how it can be used in the decision-making process, design and develop a data model to meet client requirements</p>	<p>business to meet requirements</p> <p>Examine the computational thinking skills and principles of computer programming, design and develop a software solution to meet client requirements.</p> <p>Investigate data modelling and how it can be used in the decision-making process, design and develop a data model to meet client requirements</p>	<p>business to meet requirements</p> <p>Examine the computational thinking skills and principles of computer programming, design and develop a software solution to meet client requirements.</p> <p>Investigate the principles and methodologies of IT project management as used in industry, carry out the planning, execution, monitoring and controlling of an IT project, using an appropriate methodology</p>	<p>business to meet requirements.</p> <p>Examine the computational thinking skills and principles of computer programming, design and develop a software solution to meet client requirements.</p> <p>Investigate the principles and methodologies of IT project management as used in industry, carry out the planning, execution, monitoring and controlling of an IT project, using an appropriate methodology</p>	<p>Understand the principles of website development, design and develop a website to meet client requirements.</p> <p>Investigate the principles and methodologies of IT project management as used in industry, carry out the planning, execution, monitoring and controlling of an IT project, using an appropriate methodology</p>
ASSESSMENT	<p>Written and verbal feedback provided per criteria for each unit. Unit 1 and 2 are assessed externally. Unit 1 is an exam and Unit 2 takes the form of a controlled coursework which is carried out under exam conditions. The results of Unit 1 and 2 are returned as N (Near Pass), Pass, Merit and Distinction. Unit 3 and 5 are internally assessed work graded Distinction (A), Merit(B) Pass (E). Feedback provided guides students to areas of the specification which they have not met.</p>					
USEFUL RESOURCES / GUIDANCE:						
<p>BTEC Nationals Information Technology Student Book + Active book: For the 2016 specifications (BTEC Nationals IT 2016) – ISBN – 978-1292140414)</p> <p>https://qualifications.pearson.com/en/qualifications/btec-nationals/computing-2016.html</p> <p>http://www.teach-ict.com/</p> <p>https://www.knowitallninja.com/</p>						

YEAR 13 SINGLE	AUTUMN 1	AUTUMN 2	SPRING 1	SPRING 2	SUMMER 1	SUMMER 2
CONTENT	Unit 3: Using Social Media in Business	Unit 3: Using Social Media in Business	Unit 6: Website Development	Unit 6: Website Development	Unit 6: Website Development	
SKILLS	<p>Explore the impact of social media on the ways in which businesses promote their products and services.</p> <p>Develop a plan and implement the use social media in a business to meet requirements</p>	<p>Explore the impact of social media on the ways in which businesses promote their products and services.</p> <p>Develop a plan and implement the use social media in a business to meet requirements</p>	<p>Understand and explore the principles of website development and design.</p> <p>Design and develop a website to meet client requirements.</p>	<p>Understand and explore the principles of website development and design.</p> <p>Design and develop a website to meet client requirements.</p>	<p>Understand and explore the principles of website development and design.</p> <p>Design and develop a website to meet client requirements.</p>	



Curriculum & Assessment Map

ASSESSMENT	Written and verbal feedback provided per criteria for each unit. Units 3 and 6 are internally assessed work graded Distinction (A) , Merit(B) Pass (E). Feedback provided guides students to areas of the specification which they have not met.
-------------------	--

USEFUL RESOURCES / GUIDANCE: BTEC Nationals Information Technology Student Book + Active book: For the 2016 specifications (BTEC Nationals IT 2016) – ISBN – 978-1292140414 https://qualifications.pearson.com/en/qualifications/btec-nationals/computing-2016.html http://www.teach-ict.com/ https://www.knowitallninja.com/
--

YEAR 13 TRIPLE	AUTUMN 1	AUTUMN 2	SPRING 1	SPRING 2	SUMMER 1	SUMMER 2
CONTENT	<p>Unit 11: Cyber Security and Incident Management</p> <p>Unit 14: IT Service Delivery</p> <p>Unit 8: Computer Games Development</p>	<p>Unit 11: Cyber Security and Incident Management</p> <p>Unit 14: IT Service Delivery</p> <p>Unit 8: Computer Games Development</p>	<p>Unit 11: Cyber Security and Incident Management</p> <p>Unit 14: IT Service Delivery</p> <p>Unit 8: Computer Games Development</p>	<p>Unit 17: 2D and 3D Digital Graphics</p> <p>Unit 20: Enterprise in IT</p> <p>Unit 13: Software Testing</p>	<p>Unit 17: 2D and 3D Digital Graphics</p> <p>Unit 20: Enterprise in IT</p> <p>Unit 13: Software Testing</p>	
SKILLS	<p>Demonstrate and apply knowledge and understanding of technical language, security threats, system vulnerabilities and security protection methods, and implications resulting from successful threats in order to risk assess systems and select appropriate tools to secure them</p> <p>Be able to plan a secure computer network and manage security incidents with appropriate Justification</p> <p>Demonstrate and apply knowledge and understanding of IT service-delivery related facts, terminology,</p>	<p>Demonstrate and apply knowledge and understanding of technical language, security threats, system vulnerabilities and security protection methods, and implications resulting from successful threats in order to risk assess systems and select appropriate tools to secure them</p> <p>Be able to plan a secure computer network and manage security incidents with appropriate Justification</p> <p>Demonstrate and apply knowledge and understanding of IT service-delivery related facts, terminology,</p>	<p>Demonstrate and apply knowledge and understanding of technical language, security threats, system vulnerabilities and security protection methods, and implications resulting from successful threats in order to risk assess systems and select appropriate tools to secure them</p> <p>Be able to plan a secure computer network and manage security incidents with appropriate Justification</p> <p>Demonstrate and apply knowledge and understanding of IT service-delivery related facts, terminology,</p>	<p>Investigate the purpose and characteristics of digital graphics that are an important part of visual communications</p> <p>Design 2D and 3D digital graphics products to meet a client brief using design documentation digital graphics processing and techniques</p> <p>Develop 2D and 3D digital graphics products to meet a client brief using relevant tools and techniques for developing digital graphics</p> <p>Explore the nature of enterprise and entrepreneurship in an IT context</p> <p>Develop a marketing plan</p>	<p>Investigate the purpose and characteristics of digital graphics that are an important part of visual communications</p> <p>Design 2D and 3D digital graphics products to meet a client brief using design documentation digital graphics processing and techniques</p> <p>Develop 2D and 3D digital graphics products to meet a client brief using relevant tools and techniques for developing digital graphics</p> <p>Explore the nature of enterprise and entrepreneurship in an IT context</p> <p>Develop a marketing plan</p>	



Curriculum & Assessment Map

	standards, concepts and processes explore outcomes and find solutions to IT service delivery problems. Be able to design an IT service delivery solution for an organisation with appropriate Justification.	standards, concepts and processes explore outcomes and find solutions to IT service delivery problems. Be able to design an IT service delivery solution for an organisation with appropriate Justification.	standards, concepts and processes explore outcomes and find solutions to IT service delivery problems. Be able to design an IT service delivery solution for an organisation with appropriate Justification.	for an IT product or service based on market research Present a plan for a start-up IT enterprise using lean or traditional business principles	for an IT product or service based on market research Present a plan for a start-up IT enterprise using lean or traditional business principles	
	Investigate technologies used in computer gaming Design a computer game to meet client requirements using computer games design processes and techniques Develop a computer game to meet client requirements using principles of computer games development	Investigate technologies used in computer gaming Design a computer game to meet client requirements using computer games design processes and techniques Develop a computer game to meet client requirements using principles of computer games development	Investigate technologies used in computer gaming Design a computer game to meet client requirements using computer games design processes and techniques Develop a computer game to meet client requirements using principles of computer games development	Understand the software development and testing methodologies commonly used during the development life cycle to quality assure software Carry out a range of testing methodologies on a software product to meet a client's needs Review and present the results from software tests to meet a client's needs and suggest improvements	Understand the software development and testing methodologies commonly used during the development life cycle to quality assure software Carry out a range of testing methodologies on a software product to meet a client's needs Review and present the results from software tests to meet a client's needs and suggest improvements	
ASSESSMENT	Written and verbal feedback provided per criteria for each unit. Unit 11 and 14 are assessed externally. Both units 11 and 14 are an exam are carried out under exam conditions where the results are returned as N (Near Pass), Pass, Merit and Distinction. Unit 8, 17, 20, 13 are all internally assessed work graded Distinction (A), Merit(B) Pass (E). Feedback provided guides students to areas of the specification which they have not met.					
USEFUL RESOURCES / GUIDANCE: BTEC Nationals Information Technology Student Book + Active book: For the 2016 specifications (BTEC Nationals IT 2016) – ISBN – 978-1292140414 https://qualifications.pearson.com/en/qualifications/btec-nationals/computing-2016.html http://www.teach-ict.com/ https://www.knowitallninja.com/						