



Computer Science

FOUNDATION STAGE

YEAR 7	ROTATION 1	ROTATION 2
CONTENT	Induction Emailing, internet searching, e-safety, computer components, binary Game Maker Analysing games, designing a game, using tools of Game Maker software 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 hardware components, working with binary numbers 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	Desktop publishing and design skills, numerical and spreadsheet skills, presentation and oracy skills
ASSESSMENT	Written feedback at mid-point. Assess knowledge of e-safety or computer components with a grade 1-9. Written feedback at mid- point. Assess Game Maker skills with a grade 1-9	Written feedback at mid- point. Assess game cover and presentation with a grade 1-9
USEFUL RESOURCES/GUIDANCE: https://www.thinkuknow.co.uk/ https://certificate.onlinesafetyalliance.org/ http://www.teach-ict.com/ http://www.yoyogames.com/gamemaker/windows		

YEAR 8	ROTATION 1	ROTATION 2
CONTENT	Control and Programming Understanding basic commands in programming environments such as LOGO and Scratch Using the small basic programming environment to apply syntax	Graphics and Website Understanding and creating different types of graphics Creating websites using HTML and Dreamweaver
SKILLS	Applying syntax and sequencing, programming constructs using variables, selecting appropriate controls to create animations, Logical operators, 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 vector and bitmap graphics, using the Photoshop and Illustrator software and different tools to create and edit graphics. Creating websites in HTML - adding text, images and links. Using Dreamweaver to create a website, adding CSS, applying JavaScript to manipulate code
ASSESSMENT	Written feedback at mid-point. Assess programming skills with a grade 1-9	Written feedback at mid-point. Assess graphics and website creation skills with a grade 1-9



Curriculum & Assessment Map

USEFUL RESOURCES/GUIDANCE:

<http://www.codecademy.com/>
<http://www.bbcbasic.co.uk/bbcbasic.html>
<https://www.microbit.co.uk/blocks/lessons>
<https://www.python.org/>
<http://www.teach-ict.com/>

EXAMINATION STAGE

COMPUTER SCIENCE						
YEAR 9	AUTUMN 1	AUTUMN 2	SPRING 1	SPRING 2	SUMMER 1	SUMMER 2
CONTENT	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	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	Exemplar NEA - Students will complete an exemplar of the programming project in which they will apply all the skills, knowledge and understanding of: - Decomposition and abstraction - Programming - Algorithms	Exemplar NEA - Students will complete an exemplar of the programming project in which they will apply all the skills, knowledge and understanding of: - Decomposition and abstraction - Programming - Algorithms
SKILLS	- Understand what an algorithm is, what algorithms are used for and be able to interpret algorithms (flowcharts, pseudocode, written descriptions, program code) - 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	- Be able to use sequencing, selection and iteration constructs in their programs - Understand the need for, and understand how to use, data types (integer, real, Boolean, char) - Understand the need for, and understand how to use, data structures (records, one-dimensional arrays, two-dimensional arrays)	- 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 - Be able to differentiate between types of error in programs (logic, syntax, runtime)	- 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	- Understand how to decompose a problem - 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 to create an algorithm to solve a particular problem, making use of programming constructs and using appropriate conventions (flowchart,	- Understand how to decompose a problem - 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 to create an algorithm to solve a particular problem, making use of programming constructs and using appropriate conventions (flowchart,



Curriculum & Assessment Map

	<p>description, draft program code)</p> <ul style="list-style-type: none"> - Understand the purpose of a given algorithm and how an algorithm works - Understand how to determine the correct output of an algorithm for a given set of data - Understand how to identify and correct errors in algorithms - Understand how to code an algorithm in a high-level language - Understand how the choice of algorithm is influenced by the data structures and data values that need to be manipulated - Understand how standard algorithms (bubble sort, merge sort, linear search, binary search) work - Be able to evaluate the fitness for purpose of algorithms in meeting specified requirements efficiently using logical reasoning and test data 	<ul style="list-style-type: none"> - Understand the need for, and how to manipulate, strings - Understand the need for, and how to use variables and constants - Understand the need for, and how to use, global and local variables - Understand the need for, and how to implement, validation - Be able to write code that reads/writes from/to a text file - Understand how to use, arithmetic operators (add, subtract, divide, multiply, modulus, integer division) - Understand how to use, relational operators (equal to, less than, greater than, not equal to, less than or equal to, greater than or equal to) - Understand the purpose of, and how to use, logic operators (AND, OR, NOT) 	<ul style="list-style-type: none"> - 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 - Be able to determine the strengths and weaknesses of a program and improve it - Understand the benefits of using subprograms and - Be able to write code that uses user-written and pre-existing subprograms - Understand the concept of passing data into and out of subprograms (procedures, functions) - Be able to create subprograms that use parameters 		<p>pseudo-code, written description, draft program code)</p> <ul style="list-style-type: none"> - Understand how to code an algorithm in a high-level language - Understand how the choice of algorithm is influenced by the data structures and data values that need to be manipulated - Understand how to write code that accepts and responds appropriately to user input - Be able to use sequencing, selection and iteration constructs in their programs - Understand the need for, and understand how to use, data structures - Understand the need for, and how to manipulate, strings - Understand the need for, and how to use variables and constants - Understand the need for, and how to use, global and local variables - Understand the need for, and how to implement, validation - Be able to write code that reads/writes from/to a text file 	<p>pseudo-code, written description, draft program code)</p> <ul style="list-style-type: none"> - Understand how to code an algorithm in a high-level language - Understand how the choice of algorithm is influenced by the data structures and data values that need to be manipulated - Understand how to write code that accepts and responds appropriately to user input - Be able to use sequencing, selection and iteration constructs in their programs - Understand the need for, and understand how to use, data structures - Understand the need for, and how to manipulate, strings - Understand the need for, and how to use variables and constants - Understand the need for, and how to use, global and local variables - Understand the need for, and how to implement, validation - Be able to write code that reads/writes from/to a text file
ASSESSMENT	Written feedback half termly. Assessed work graded 1-9 (formative)	Written feedback half termly. Assessed work graded 1-9 (formative)	Written feedback half termly. Assessed work graded 1-9 (formative)	Written feedback half termly. Assessed work graded 1-9 (formative)	Written feedback half termly. Assessed work graded 1-9 (formative)	Written feedback half termly. Assessed work graded 1-9 (formative)
USEFUL RESOURCES/GUIDANCE:						



Curriculum & Assessment Map

COMPUTER SCIENCE						
YEAR 10	AUTUMN 1	AUTUMN 2	SPRING 1	SPRING 2	SUMMER 1	SUMMER 2
CONTENT	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 Data - Binary - Data representation	Programming - Understand the structural components of a program (variable and type declarations, command sequences, selection, iteration, data structures, subprograms) Data - Data representation - Data storage and compression - Encryption	Programming - Understand how to and be able write programs in a high-level programming language - Understand the, need use and benefits of using subprograms Data and Computers - Databases - Machines and computational models	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 Computers - Computer Hardware - Logic	Exemplar NEA Students will complete an exemplar of the programming project in which they will apply all the skills, knowledge and understanding of: -Decomposition and abstraction -Programming -Algorithms Software - Systems Software - Applications Software - Programming languages	NEA Students will start the programming project (NEA) in which they will apply all the skills, knowledge and understanding of: -Decomposition and abstraction -Programming -Algorithms
SKILLS	- 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 that computers use binary to represent data	- 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 how bitmap images are represented in binary	- 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 - Be able to interpret error messages and identify, locate and fix errors in a program	- 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 -understand the function of different	Application of algorithms, programming and decomposition knowledge and skills to complete exemplar NEA - Know what an operating system its functions - Understand the purpose and functions of utility software (managing, repairing and converting files; compression;	Application of algorithms, programming and decomposition knowledge and skills to NEA "programming project"

- <https://qualifications.pearson.com/en/qualifications/edexcel-gcses/computer-science-2016.html>
- <http://www.codecademy.com/>
- <https://www.python.org/>
- <http://www.teach-ict.com/>
- <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

	<p>(numbers, text, sound, graphics) and program instructions</p> <ul style="list-style-type: none"> - 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 how computers encode characters [ASCII, Unicode] - Understand how bitmap images are represented in binary [pixels, resolution, colour depth] - Understand how analogue data [sound, temperature, light intensity] is represented in binary 	<p>[pixels, resolution, colour depth]</p> <ul style="list-style-type: none"> - Understand how analogue data [sound, temperature, light intensity] is represented in binary - 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 	<ul style="list-style-type: none"> - Understand the benefits of using subprograms <p>Understand the characteristics of 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>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] 	<p>defragmentation; backing up; anti-virus, anti-spyware)</p> <ul style="list-style-type: none"> - 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 	
ASSESSMENT	Written feedback half termly. Assessed work graded 1-9 (formative)	Written feedback half termly. Assessed work graded 1-9 (formative)	Written feedback half termly. Assessed work graded 1-9 (formative)	Written feedback half termly. Assessed work graded 1-9 (formative)	Written feedback half termly. Assessed work graded 1-9 (formative)	Written feedback half termly. Assessed work graded 1-9 (formative)
USEFUL RESOURCES/GUIDANCE:						



Curriculum & Assessment Map

- <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/UC0HzEBLIJxlrwBAHJ5S9JQg>
- <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

COMPUTER SCIENCE						
YEAR 11	AUTUMN 1	AUTUMN 2	SPRING 1	SPRING 2	SUMMER 1	SUMMER 2
CONTENT	Networks - Networking - Types of networks - Connectivity and speed - Topologies - Transmission and protocols - Network Security Revisiting topic - Data - Binary - Data representation - Data storage and compression - Encryption	The internet and the world wide web - Internet structure - WWW and its components - HTML and CSS - Server side processing Revisiting topic - Data and Computers - Databases - Machines and computational models	Revisiting topic - 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 Revisiting topic - Computers and Software - Computer Hardware - Logic - Systems Software - Applications Software - Programming languages	Revisiting topic - Programming - 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 The Bigger Picture - Emerging trends, issues and impact	Revision	Study Leave
SKILLS	- 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,	- 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,	- 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)	- 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 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	



Curriculum & Assessment Map

<p>Gbps] - understand the role of and need for network protocols</p> <ul style="list-style-type: none"> - Understand that data can be transmitted over networks [TCP/IP] - Understand characteristics of network topologies - understand the importance of network security and methods of identifying vulnerabilities <p>- Understand that computers use binary to represent data (numbers, text, sound, graphics) and program instructions</p> <ul style="list-style-type: none"> - 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), 	<p>links, images, media, layout, styles, lists]</p> <ul style="list-style-type: none"> - Understand the client-server model, the difference between client-side and server-side processing and the role of cookies <p>Understand the characteristics of 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 	<ul style="list-style-type: none"> - 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 - understand the function of different types of main memory (RAM, ROM, cache) - 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; 	<ul style="list-style-type: none"> - 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 - 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] 	<p>aspects of the real world</p> <ul style="list-style-type: none"> - Application of algorithms and programming knowledge to exam questions - Application of theory of computer science topics to a variety of questions 	
--	---	--	---	--	--



Curriculum & Assessment Map

	<p>gigabyte (GB), terabyte (TB)</p> <ul style="list-style-type: none"> - 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>defragmentation; backing up; anti-virus, anti-spyware)</p> <ul style="list-style-type: none"> - 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 			
ASSESSMENT	Written feedback half termly. Assessed work graded 1-9 (formative)	Written feedback half termly. Assessed work graded 1-9 (formative)	Written feedback half termly. Assessed work graded 1-9 (formative)	Written feedback half termly. Assessed work graded 1-9 (formative)	Written feedback half termly. Assessed work graded 1-9 (formative)	Written feedback half termly. Assessed work graded 1-9 (formative)
<p>USEFUL RESOURCES/GUIDANCE:</p> <ul style="list-style-type: none"> • https://qualifications.pearson.com/en/qualifications/edexcel-gcses/computer-science-2016.html • http://www.codecademy.com/ • https://www.python.org/ • http://www.teach-ict.com/ • https://www.youtube.com/channel/UC0HzEBLIJxlrwBAHJ5S9JQg • 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

GRAPHIC COMMUNICATION						
YEAR 9	AUTUMN 1	AUTUMN 2	SPRING 1	SPRING 2	SUMMER 1	SUMMER 2
CONTENT	Skills Development - Photoshop	Skills Development - Photoshop	Skills Development - Illustrator	Component 1: Coursework (60% of total GCSE grade) Ice Cream Shop Logo research, developing & creating	Component 1: Coursework (60% of total GCSE grade) Ice Cream Shop Logo research, developing & creating	Component 1: Coursework (60% of total GCSE grade) Movie Promotion project. Artist research, developing, creating & recording
SKILLS	Experimentation with a variety of software tools, observational/recording skills. Learning and using the tools of Photoshop to create, adapt and refine graphics.	Experimentation with a variety of software tools, observational/recording skills. Learning and using the tools of Photoshop to create, adapt and refine graphics. Applying skills and knowledge of Photoshop to Promotional Poster Project	Experimentation with a variety of software tools, observational/recording skills. Learning and using the tools of Illustrator to create, adapt and refine graphics. Applying skills and knowledge of Illustrator to Company Logo Project	Artist research including annotations, notes, transcriptions. Developing ideas making links to artist research, existing graphics and theme of work.	Review and modify own work, reflection and annotation of own work. Create a personal outcome in graphics and typography- using tools and skills in Photoshop and/or Illustrator. Use of colour, stylisation, scale and composition	Artist research including annotations, notes, transcriptions. Developing ideas making links to artist research, existing graphics and theme of work.
ASSESSMENT	Written feedback half termly. Assessed work graded 1-9 (formative)	Written feedback half termly. Assessed work graded 1-9 (formative)	Written feedback half termly. Assessed work graded 1-9 (formative)	Written feedback half termly. Assessed work graded 1-9 (formative)	Written feedback half termly. Assessed work graded 1-9 (formative)	Written feedback half termly. Assessed work graded 1-9 (formative)
USEFUL RESOURCES/GUIDANCE: http://www.aqa.org.uk/subjects/art-and-design/gcse/art-and-design-8201-8206						



Curriculum & Assessment Map

GRAPHIC COMMUNICATION						
YEAR 10	AUTUMN 1	AUTUMN 2	SPRING 1	SPRING 2	SUMMER 1	SUMMER 2
CONTENT	Component 1: Coursework (60% of total GCSE grade) Movie Promotion project Artist research, developing, creating & recording	Component 1: Coursework (60% of total GCSE grade) Movie Promotion project Artist research, developing, creating & recording	Component 1: Coursework (60% of total GCSE grade) Movie Promotion project Artist research, developing, creating & recording	Component 1: Coursework (60% of total GCSE grade) Festival Project(advertising products) Artist research, developing, creating & recording	Component 1: Coursework (60% of total GCSE grade) Festival Project(advertising products) Artist research, developing, creating & recording	Component 1: Coursework (60% of total GCSE grade) Festival Project(advertising products) Artist research, developing, creating & recording
SKILLS	Artist research including annotations, notes, transcriptions. Developing ideas making links to artist research, existing graphics and theme of work.	Review and modify own work, reflection and annotation of own work. Create different outcomes using tools, techniques.	Create a personal outcome in graphics and typography- using tools and skills in Photoshop and/or Illustrator. Use of colour, stylisation, scale and composition	Artist research including annotations, notes, transcriptions. Developing ideas making links to artist research, consider intended audience, how graphics are used to communicate ideas and theme of work. Recording from primary and secondary sources. Experiment with a variety of media. Students should be growing in independence and selecting their own resources and materials	Review and modify own work, reflection and annotation of own work reflecting on particular aim and purpose/mood. Create a personal outcome in graphics and typography- using tools and skills in Photoshop and/or Illustrator. Use of pattern, typography, audience, purpose.	Review and modify own work, reflection and annotation of own work reflecting on particular aim and purpose/mood. Create a personal outcome in graphics and typography- using tools and skills in Photoshop and/or Illustrator. Use of pattern, typography, audience, purpose.
ASSESSMENT	Written feedback half termly. Assessed work graded 1-9 (formative)	Written feedback half termly. Assessed work graded 1-9 (formative)	Written feedback half termly. Assessed work graded 1-9 (formative)	Written feedback half termly. Assessed work graded 1-9 (formative)	Written feedback half termly. Assessed work graded 1-9 (formative)	Written feedback half termly. Assessed work graded 1-9 (formative)
USEFUL RESOURCES/GUIDANCE: http://www.aqa.org.uk/subjects/art-and-design/gcse/art-and-design-8201-8206						



Curriculum & Assessment Map

GRAPHIC COMMUNICATION						
YEAR 11	AUTUMN 1	AUTUMN 2	SPRING 1	SPRING 2	SUMMER 1	SUMMER 2
CONTENT	Component 1: Coursework (60% of total GCSE grade) Festival Project(advertising products) Artist research, developing, creating & recording	Creating a portfolio of coursework- 60% of total GCSE grade.	Component 2: GRAPHICAL COMMUNICATION EXAM Students select their own stating point from a selection of questions set externally.	Component 2: GRAPHICAL COMMUNICATION EXAM Development.	Component 2: GRAPHICAL COMMUNICATION EXAM Creating outcome.	
SKILLS	Create a personal outcome in graphics and typography- using tools and skills in Photoshop and/or Illustrator. Use of pattern, typography, audience, purpose.	Review, modify and refine all Component 1 coursework. Create a formal portfolio that demonstrates all four- assessment objectives- recording, experimenting, developing, outcome.	Artist research including annotations, notes, transcriptions. Recording from primary and secondary sources. Experiment with a variety of media. Students should be working independently and selecting their own resources and materials.	Developing personal and meaningful ideas and responses, making links to chosen artist. Developing skills in their chosen media.	Review modify and refine ideas. 10-hour practical exam over two days. Students are to create their final outcome and submit all the preparatory work at the beginning of the exam.	
ASSESSMENT	Written feedback half termly. Assessed work graded 1-9 (formative)	All coursework is internally marked and externally modified. Official coursework grade (60% of total GCSE grade)	Written feedback half termly. Assessed work graded 1-9 (formative)	Written feedback half termly. Assessed work graded 1-9 (formative)	All exam work is internally marked and externally modified. Official exam grade. (40% of total GCSE grade)	
USEFUL RESOURCES/GUIDANCE:						



Curriculum & Assessment Map

ADVANCED STAGE

AS LEVEL COMPUTER SCIENCE						
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 and 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), 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 Practical Work: Write code in python using data structures to solve problem. Write algorithms to solve simple problems.	Record data types 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 positive integers between Binary Hexadecimal and denary. Positive and negative real numbers using normalised floating point representation. How character sets (ASCII and UNICODE) are used to represent text. Record and study the laws for data protection, data misuse, copyright and	Develop knowledge and understanding of Legal, moral, ethical and cultural issues. Develop deeper knowledge and understanding of: 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	Develop knowledge and understanding of: Understand the 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	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. Make a start on the project report. 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

			investigatory powers. Record standard algorithms for searching and sorting. Record standard algorithms for add, delete items from stack and queue. Activity on the suitability of algorithms for solving a problem. Develop knowledge and understanding of: Thinking abstractly and Thinking ahead	knowledge and understanding of Lossy v lossless compression.	source vs Closed source. Translators: interpreters, compilers and assemblers. Develop knowledge and understanding of: Boolean expressions, Karnaugh maps.	
ASSESSMENT	Written feedback half termly. Assessed work graded A-E (formative)	Written feedback half termly. Assessed work graded A-E (formative)	Written feedback half termly. Assessed work graded A-E (formative)	Written feedback half termly. Assessed work graded A-E (formative)	Written feedback half termly. Assessed work graded A-E (formative)	Written feedback half termly. Assessed work graded A-E (formative)
USEFUL RESOURCES/GUIDANCE: <ul style="list-style-type: none"> • 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://www.youtube.com/channel/UC0HzEBLIJxlrwBAHJ5S9JQg • 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) 						



Curriculum & Assessment Map

A LEVEL COMPUTER SCIENCE						
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	Study Leave
SKILLS	Develop knowledge and understanding of: Representation and normalisation of floating point numbers in binary. 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. Develop knowledge and understanding of: Normalisation to 3NF SQL – Interpret and modify. Referential integrity. Transaction processing, ACID Define the stakeholder for the problem. Make a start on the project report to explain the type of game, define the user requirement and	Complete the project report on analysis. Decompose the game development problem, then explain and justify the structure of the solution. Describe the parts of the solution using algorithms justifying how these algorithms form a complete solution to the problem. Develop knowledge and understanding of: The suitability of different algorithms for a given task and data set, in terms of execution time and space. Develop knowledge and understanding of: The following structures to store data: linked-list, graph (directed and undirected). Develop coding in Monkey-x to develop the game. Obtain feedback from the stakeholders,	Develop knowledge and understanding of: The nature, benefits and drawbacks of caching. Develop knowledge and understanding of: The suitability of different algorithms for a given task and data set, in terms of execution time and space. 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. Develop knowledge and understanding of: The	Develop knowledge and understanding of: Measures and methods to determine the 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) Develop knowledge and understanding of: The nature, benefits and drawbacks of caching. Develop knowledge and understanding of: How to create, traverse, add data to and remove data from the data structures mentioned above. Develop knowledge and understanding of: Search engine indexing. PageRank algorithm. Server and client side	Develop knowledge and understanding of: Determine the parts of a problem that can be tackled at the same time. Outline the benefits and trade offs that might result from concurrent processing in a particular situation. Develop knowledge and understanding of: Recursion, how it can be used and compares to an iterative approach. Develop knowledge and understanding of: Features that make a problem solvable by computational methods. Problem recognition. Problem decomposition. Use of divide and conquer. Use of abstraction. Apply knowledge of: <ul style="list-style-type: none"> • backtracking • data mining • heuristics 	Study Leave



Curriculum & Assessment Map

	<p>stake holders. Develop knowledge and understanding of: Procedural languages. Assembly language Modes of addressing memory Develop knowledge and understanding of: Stages of compilation Linkers, Loaders and the use of libraries Develop knowledge and understanding of: The use of pipelining in a processor to improve efficiency GPUs and their uses Develop knowledge and understanding of: Use the following rules to derive or simplify statements in Boolean algebra: De Morgan's Laws, distribution, association, commutation, double negation. The logic associated with D type flip flops, half and full adders. Students work on writing the code in Monkey-x to develop their game. Progress recorded in report.</p>	<p>analyse their suggestions and make improvement on the coding. Create test plan and record evidence of tests.</p>	<p>following structures to store data: stack, queue, tree, binary search tree, hash table. 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>processing. Develop knowledge and understanding of: Network security and threats, use of firewalls, proxies and encryption. Network hardware.</p>	<ul style="list-style-type: none"> • performance modelling • pipelining • visualisation to solve problems. <p>Develop knowledge and understanding of: Object oriented techniques. Develop knowledge and understanding of: Run length encoding and dictionary coding for lossless compression. Symmetric and asymmetric encryption. Different uses of hashing.</p>	
ASSESSMENT	Written feedback half termly. Assessed work graded A-E (formative)	Written feedback half termly. Assessed work graded A-E (formative)	Written feedback half termly. Assessed work graded A-E (formative)	Written feedback half termly. Assessed work graded A-E (formative)	Written feedback half termly. Assessed work graded A-E (formative)	Study Leave
<p>USEFUL RESOURCES/GUIDANCE:</p> <ul style="list-style-type: none"> • 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://www.youtube.com/channel/UC0HzEBLIJxlrwBAHJ5S9JQg • 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) 						



Curriculum & Assessment Map

YEAR 12 BTEC IT – Single NEW SPEC	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/						

YEAR 12 BTEC IT – Triple NEW SPEC	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 9: IT Project Management	Unit 3: Using Social Media in Business Unit 4: Programming Unit 9: IT Project Management	Unit 3: Using Social Media in Business Unit 6: Website Development	Unit 3: Using Social Media in Business Unit 6: Website Development
SKILLS	Demonstrate knowledge and understanding of	Demonstrate knowledge and understanding of	Demonstrate knowledge and understanding of	Explore the impact of social media on the ways in which	Explore the impact of social media on the ways in which	Explore the impact of social media on the ways in which



Curriculum & Assessment Map

	<p>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</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>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</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>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</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>businesses promote their products and services. Develop a plan and implement the use social media in a 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>businesses promote their products and services. Develop a plan and implement the use social media in a business to meet requirements.</p> <p>Understand the principles of website development, design and develop a website to meet client requirements.</p>	<p>businesses promote their products and services. Develop a plan and implement the use social media in a business to meet requirements.</p> <p>Understand the principles of website development, design and develop a website to meet client requirements.</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 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.</p>					
<p>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/</p>						



Curriculum & Assessment Map

YEAR 13 BTEC IT - Single LEGACY	AUTUMN 1	AUTUMN 2	SPRING 1	SPRING 2	SUMMER 1
CONTENT	Unit 22 – Developing Computer Games	Unit 11 - Systems analysis & Design	Unit 11 - Systems analysis & Design	Unit 9 – Computer Networks	Unit 9 – Computer Networks
SKILLS	<p>Research - examining the psychological effects of computer gaming on individuals and society.</p> <p>Programming – design and produce a computer game using GML.</p> <p>Evaluative - suggesting improvements to a computer game following user feedback</p>	<p>Analysis – Looking at an existing business's system to identify improvements.</p> <p>Design – Design a potential improvement to the existing system</p> <p>Research – Research possible solutions to existing problems in an organisation.</p>	<p>Analysis – Looking at an existing business's system to identify improvements.</p> <p>Design – Design a potential improvement to the existing system</p> <p>Research – Research possible solutions to existing problems in an organisation.</p>	<p>Troubleshooting skills – exploring the issues, events or problems involved with peer-to-peer network and client/server networks, from different perspectives</p>	<p>Troubleshooting skills – exploring the issues, events or problems involved with peer-to-peer network and client/server networks, from different perspectives</p>
ASSESSMENT	Written and verbal feedback provided per criteria for each unit. Assessed work graded Distinction (A) to Pass (E). Feedback provided guides students to areas of the specification which they have not met.				
USEFUL RESOURCES/GUIDANCE: BTEC Level 3 National IT Student Book 1 (BTEC National for IT Practitioners) – ISBN – 1846909287 BTEC Level 3 National IT Student Book 2 (BTEC National for IT Practitioners) - ISBN 1846909295 https://qualifications.pearson.com/en/qualifications/btec-nationals/it-2010.html http://www.teach-ict.com/					

YEAR 13 BTEC IT - Triple LEGACY	AUTUMN 1	AUTUMN 2	SPRING 1	SPRING 2	SUMMER 1
CONTENT	Unit 11 - Systems Analysis & Design <i>Unit 16 - Procedural Programming</i>	Unit 31 – Computer Animation <i>Unit 15 - Object Orientated Programming</i>	Unit 8 – E-Commerce <i>Unit 26 - Maths for IT Practitioners</i>	Unit 8 – E-Commerce <i>Unit 22 – Developing Computer Games</i>	Unit 3 – Information Systems <i>Unit 43 – Multimedia Design</i>
SKILLS	<p>Analysis – Looking at an existing business's system to identify improvements.</p> <p>Design – Design a potential improvement to the existing system.</p> <p><i>Programming - generating ideas and exploring</i></p>	<p>Research - Describing factors that need to be taken into account when creating animations for the web.</p> <p>Creative - designing computer animations using different animation techniques.</p>	<p>Independent thinking - exploring issues, events or problems e-commerce causes for society from different perspectives</p> <p>Creative Thinking - connecting their own and others' ideas and experiences in inventive</p>	<p>Independent thinking - exploring issues, events or problems e-commerce causes for society from different perspectives</p> <p>Creative Thinking - connecting their own and others' ideas and experiences in inventive</p>	<p>Analysis - analysing and evaluating information, judging its relevance and value when using IT tools to produce management information.</p> <p>Evaluative - supporting conclusions, using reasoned arguments and evidence when justifying</p>



Curriculum & Assessment Map

	<p><i>possibilities to design and code a procedural program.</i></p> <p><i>Communication - communicating their learning by writing support documentation for a procedural program.</i></p>	<p><i>Programming - working towards goals, showing initiative, commitment and perseverance when developing an object oriented application</i></p> <p><i>Planning - anticipating, taking and managing risks when testing an object oriented application.</i></p>	<p>ways to produce an e-commerce strategy.</p> <p><i>Mathematical and Analytical skills - analysing and evaluating information, interpreting trends in data.</i></p> <p><i>Reflective learners - reviewing progress, acting on the outcomes of the results of a study.</i></p>	<p>ways to produce an e-commerce strategy.</p> <p><i>Research - examining the psychological effects of computer gaming on individuals and society.</i></p> <p><i>Programming - design and produce a computer game using GML.</i></p>	<p>information selection.</p> <p><i>Reflective learners - inviting feedback on their own work and dealing positively with praise, setbacks and criticism.</i></p> <p><i>Organisation - organising time and resources and prioritising actions when planning to produce a multimedia product, whether working on their own or in a group.</i></p>
ASSESSMENT	Written and verbal feedback provided per criteria for each unit. Assessed work graded Distinction (A) to Pass (E). Feedback provided guides students to areas of the specification which they have not met.				
<p>USEFUL RESOURCES/GUIDANCE:</p> <p>BTEC Level 3 National IT Student Book 1 (BTEC National for IT Practitioners) - ISBN - 1846909287</p> <p>BTEC Level 3 National IT Student Book 2 (BTEC National for IT Practitioners) - ISBN 1846909295</p> <p>https://qualifications.pearson.com/en/qualifications/btec-nationals/it-2010.html</p> <p>http://www.teach-ict.com/</p>					