Year 12 OCR	Curriculum Intent: The aims of this qualification are to enable learners to develop:					
Computer Science A-Level	<ul> <li>An understanding and ability to apply the fundamental principles and concepts of computer science, including: abstraction, decomposition, logic, algorithms and data representation</li> <li>The ability to analyse problems in computational terms through practical experience of solving such problems, including writing programs to do so</li> <li>The capacity to think creatively, innovatively, analytically, logically, and critically</li> </ul>					
	<ul> <li>The capacity to see relationships between different aspects of computer science</li> </ul>					
Year 12						
Computer systems Component 1	Term 1:	Term 2:	Term 3			
Topic Titles (in order of delivery)	Paper 1: Computer Systems The characteristics of contemporary processors, input, output and storage devices Software and software development Paper 2: Programming Computational Thinking • Thinking abstractly • Thinking ahead • Think procedurally • Thinking logically / concurrently Fundamentals of C# • Use of IDE (Visual Studio 2022) • Different data types • Basic Console programs • Commenting and debugging	<ul> <li>Paper 1: Computer Systems</li> <li>Exchanging data</li> <li>Paper 2 Programming:</li> <li>Data types, data structures and algorithms</li> <li>Programming paradigms <ul> <li>Procedural</li> <li>Declarative</li> <li>Object Oriented</li> <li>Event Driven</li> </ul> </li> </ul>	Paper 1: Computer Systems Legal, moral, cultural, and ethical issues Standard Algorithms • Linear Search • Binary Search • Bubble and Insertion Sort • Merge and quick sort • Optimisation – Dijkstra / A* algorithms Paper 2: Programming Functions Analyse and define algorithms to solve problems NEA: Project Proposals Analysis			
Key knowledge / Retrieval topics	Paper 1: Computer Systems Components of a computer and their uses (a) The Arithmetic and Logic Unit; ALU, Control Unit and Registers (Program Counter; PC, Accumulator; ACC, Memory Address Register; MAR, Memory Data Register; MDR, Current Instruction Register; CIR). Buses: data, address and control: how this relates to assembly language programs. (b) The Fetch-Decode-Execute Cycle; including its effects on registers. (c) The factors affecting the performance of the CPU: clock speed number of cores	<ul> <li>Paper 1: Computer Systems</li> <li>How data is exchanged between different systems</li> <li>(a) Lossy vs Lossless compression.</li> <li>(b) Run length encoding and dictionary coding for lossless compression.</li> <li>(c) Symmetric and asymmetric encryption.</li> <li>(d) Different uses of hashing.</li> <li>Networks: <ul> <li>a) Characteristics of networks and the importance of protocols and standards.</li> <li>(b) The internet structure: <ul> <li>The TCP/IP Stack.</li> <li>DNS</li> <li>Protocol layering.</li> <li>LANs and WANs.</li> </ul> </li> </ul></li></ul>	<ul> <li>Paper 1: Computer Systems</li> <li>The individual moral, social, ethical, and cultural opportunities, and risks of digital technology.</li> <li>Legislation surrounding the use of computers and ethical issues that can or may in the future arise from the use of computers.</li> <li>Computers in the workforce.</li> <li>Automated decision making.</li> <li>Artificial intelligence.</li> <li>Environmental effects.</li> <li>Censorship and the Internet.</li> <li>Monitor behaviour.</li> <li>Analyse personal information.</li> <li>Piracy and offensive communications.</li> </ul>			

<ul> <li> <ul> <li>eache.</li> <li>Packet and dicture switching.</li> <li>Packet and dicture switching.</li> <li>Packet and dicture switching.</li> <li>Post of pipeling in a processor to improve efficiency.</li> <li>Post of pipeling in a processor to improve efficiency.</li> <li>Post of pipeling in a processor to improve efficiency.</li> <li>Post of pipeling in a processor to diverse and the different methodologies used to develop software end the different methodologies used to develop software end the different methodologies used to develop software end the different agorithms that can be applied to these structures.</li> <li>Operators</li> <li>Basic C instructions</li> <li>Operators</li> <li>Sequence / Selection / Iteration</li> <li>Track of pipeling in a processor input systems</li> <li>Structures. Diversion, Encryption and Hashing</li> <li>Trees</li> </ul> </li> <li>Understandig / Repart 1: Computer Systems</li> <li>Structures and function of the processor input systems</li> <li>Software Development</li> <li>Types of Programming Pager 2: Programming</li></ul>	-			
(d) The use of pipelining in a processor improve efficiency.       Sets.         (e) Von Neumann, Harvard and contemporar processor architecture       and encryption.       (d) Network hardware.       Standard algorithms         (e) Dies of software and the different methodologies used to develop software       Paper 2: Programming       Paper 2: Programming         Variables       Operators       Basic Chinstructions       Eable to analyse a problem         Sequence / Selection / Iteration       - Aray: (Tuples / records       Be able to compare different algorithms in terms         Structure and function of the processor       - Aray: (Tuples / records       Be able to compare different algorithms in terms         Structure and function of the processor       - Traves (Tuples / records       - Compare different algorithms to solve specific proposal         Import of project proposal       - Traves (Tuples / records       - Completion of Analysis         Structure and function of the processor       - Traves (Tuples / records       - Completion of Analysis         Import of project proposal       Completion of Analysis       Completion of Analysis         Structure and function of the processor       - Web records       - Constant functions         Types of processor       - Paper 2: Programming       - Compared algorithms to solve a problem         Software Development       - Software Development       - Compared functions		• cache.	<ul> <li>Packet and circuit switching.</li> </ul>	<ul> <li>Layout, colour paradigms and character</li> </ul>
understanding /       end encryption.       (d) Network hardware.       (e) Von Neumann, Harvard and contemporar processor architecture       (d) Network hardware.       (e) Network hardware.       (f) Network hardware. <th></th> <th>(d) The use of pipelining in a processor to</th> <th>(c) Network security and threats, use of firewalls, proxies</th> <th>sets.</th>		(d) The use of pipelining in a processor to	(c) Network security and threats, use of firewalls, proxies	sets.
(e) Von Neumann, Harvard and contemporary processor architecture       (d) Network hardware.       • Recognise, use, and trace each algorithm (e) (Client-server and peer to peer.         Types of software and the different methodologies used to develop software Deper 2: Programming       • Paper 2: Programming         Variables       Operators         Basic C# instructions       • Sequence / Selection / Iteration         Sequence / Selection / Iteration       • Analysis / Linkel Usts         • Types of processor       • Stacks         • Types of processor       • Computer Systems         Types of processor       Paper 1: Computer Systems         Software Development       Compression         Software Development       Comparison         Software Development       Stacks         Software Development       Paper 1: Computer Systems         Software Development       Compression         Software Development       Paper 2: Programming         Abstraction       Software Development         Software Development       Nager 1: Computer Systems         Software Development       Software Development         Yee of rogramming Abstraction       Software Development         Software Development       Software Development         (i) The nature of abstraction.       Software Development         (i		improve efficiency.	and encryption.	Standard algorithms
understanding/ Sequence / delivery         Paper 1: Computer Systems Structures and survey of the structures Departs         Paper 1: Computer Systems Sequence / Selection / Iteration         Paper 1: Computer Systems Sequence / Selection / Iteration         Paper 1: Computer Systems Sequence / Selection / Iteration         Paper 1: Computer Systems Structures and function of the processor Types of processor Types of processor Systems Software and buttot of the processor Types of processor Thinking shead (a) Identify the inputs and outputs for a given Thinking shead (b) D the need for assiztation. (b) D the need for assiztation. (c) The nature, benefits and drawbacks of caching. (d) The need for crusable program components. Thinking shead (a) Identify the inputs and outputs for a given Thinking shead (b) Identify the inputs an		(e) Von Neumann, Harvard and contemporary	(d) Network hardware.	Recognise, use, and trace each algorithm
Understanding/ Sequence of delivery     Paper 1: Programming How data is represented and stored within different structures. Different algorithms that can be applied to these structures. Different algorithms that can be applied to structures. Different algorithms that can be applied to structures. Different algorithms that can be applied to these structures. Different algorithms to solve specific problems     Be able to achyse a problem Be able to achyse a problem be able to achyse a problem of Big O: Time efficiency       Paper 1: Computer Systems     - Time efficiency     - Time efficiency       Paper 1: Computer Systems     - Graphs     - Completion of Analysis       Structure and function of the processor Types of processor Types of processor Types of Programming Systems Software Applications Generation. Types of Programming Abstraction     - Paper 1: Computer Systems Competion Analysis     - Paper 1: Computer Systems Completion of Analysis       (d) Devise an abstract model for a variety of situation.     - Structures     - Constant time Data Types     - Constant time Data Structures       (d) Devise an abstract model for a variety of situation.     - Structures     - Constant time Data Structures     - Constant time Data Structures       (d) Devise an abstract model for a variety of situation.     - C		processor architecture	(e) Client-server and peer to peer.	Explain usage
Image: Description         Paper 2: Programming         Paper 3: Pr		Types of software and the different	Paper 2: Programming	Paper 2: Programming
Paper 1: Programming Variables Operators Basic Cit instructions Sequence / Selection / Iteration         structures. Different algorithms that can be applied to these structures • Arrays / Tuples / records • Queues • Queues • Units / Linked Lits • Stacks • Graphs         Be able to develop algorithms to solve specific problems • Be able to develop algorithms in terms of Big O: • Time efficiency • Space efficiency • Stacks • Graphs           Paper 1: Computer Systems Structure and function of the processor Tupes of processor Input, output and stronge Systems Software Applications Generation Botices of Brogramming Abstraction (a) The nature of abstraction. (b) The need for avariety of situations. Thinking ahead (a) (defitive the preconditions for a given situation.         Paper 1: Computer Systems Completion of Analysis Paper 2: Programming Subroutines and recursion Data Structures • Arrays / Tuples / records • Lines • Arrays / Tuples / records • Li		methodologies used to develop software	How data is represented and stored within different	Be able to analyse a problem
Understanding / genered of delivery       Variables       these structures       problems         Understanding / genered delivers       Sequence / Selection / Iteration       these structures       problems         Basic C# instructions       Sequence / Selection / Iteration       problems       Be able to compare different algorithms in terms of Big O:         Paper 1: Computer Systems       Graphs       Completion of Analysis       These         Paper 1: Computer Systems       Faper 1: Computer Systems       Paper 1: Computer Systems       Paper 1: Computer Systems         Structure and function of the processor       Types of processor       Databases       Paper 1: Computer Systems         Software Development       Software Development       Bolcan Algebra       Data Types / processor         Types of processor       Data Types / programming       Subructures and recursion       Data Types / records         (b) The net me for abstraction.       (b) The net me for abstraction.       Outputes systems       Outputes systems         (c) The differences between an abstraction and reality.       (c) The differences between an abstraction and reality.       Iteras       Iteras         (a) Identify the inputs and outputs for a given struction.       (b) Determine the preconditions for devising a solution to a problem.       Frees       NEA: Requirements of a project proposal Requirements of Analysis         (d)		Paper 2: Programming	structures. Different algorithms that can be applied to	Be able to develop algorithms to solve specific
Understanding/ Believery     Operators Basic C# instructions Sequence / Selection / Iteration <ul> <li>Arrays / Tuples / records</li> <li>Basic C# instructions Sequence / Selection / Iteration</li> <li>Hash tables</li> <li>Hash tables</li> <li>Graphs</li> </ul> <ul> <li>Trees</li> </ul> <ul> <li>Paper 1: Computer Systems</li> <li>Structure and function of the processor</li> <li>Trees</li> </ul> <ul> <li>Paper 1: Computer Systems</li> <li>Structure and function of the processor</li> <li>Trypes of processor</li> <li>Input, output and storage</li> <li>Systems Software</li> <li>Boolean Algebra</li> <li>Development</li> <li>Paper 2: Programming</li> <li>Subroutines and recursion</li> <li>Boolean Algebra</li> <li>Development</li> <li>Types of Programming</li> <li>Subroutines and recursion</li> <li>Data Structures</li> <li>Arrays / Tuples / records</li> <li>Outous</li> <li>Compating algorithms</li> <li>Compating algorithms</li> <li>Constant time</li> <li>Liss / Linked Lists</li> <li>Constant import</li> <li>Polyn</li></ul>		Variables	these structures	problems
Understanding, delivery     Basic C# instructions Sequence / Selection / Iteration     • Queues • Queues • Queues • Lists / Linked Lists • Hash tables • Trees     of Big O: • Time efficiency • Space efficiency • NRA: Completion of project proposal Completion of Analysis       Paper 1: Computer Systems Structure and function of the processor Types of processor Input, output and storage Systems Software Applications Generation Software Development Types of Programming Abstraction delivery     Paper 1: Computer Systems Computer Systems Completion of Analysis     Paper 1: Computer Systems Completion of analysis       Understanding, delivery     Paper 1: Computer Systems Software Development Types of Programming Abstraction. (c) The need for abstraction. (c) The need for abstraction and reality. (c) The differences between an abstraction and reality. (c) The need for reusable program (c) Optoermine the preconditions for devising a solution to a problem. (c) The nature, benefits and drawbacks of caching. (d) The need for reusable program components. Thinking anead (c) C) The nature, benefits and drawbacks of caching. (d) The need for reusable program components. Thinking receiver and the reusable program components. Thinking the starter of the NEA Requirements of Analysis		Operators	Arrays / Tuples / records	Be able to compare different algorithms in terms
Sequence / Selection / Iteration       • Litts / Linked Lists       • Time efficiency         • Stacks       • Space efficiency         • Hash tables       • Graphs         • Organis       Completion of project proposal         • Trees       Completion of project proposal         • Trees       Completion of nalysis         • Trees       Paper 1: Computer Systems         • Trees       Paper 1: Computer Systems         • Trees       Paper 2: Programming         • Software Development       Paper 2: Programming         • Stacks       • Stacks         • Outerstanding /       • Constant time         • Sequence of delivery       (a) The need for abstraction.         (b) The need for abstraction.       • Trees         (c) The differences Detween an abstraction and reality.       • Stacks         • Graphs       • Constant time         • Graphs       • Constant time         • Graphs       • Constant time         • Constant time       • Linear         • Outers       • Graphs         • Constant time       • Linear         • Outers       • Graphs         • Graphs       • Graphs         • Constant time       • Linear         • Outers       • Graphs <th></th> <th>Basic C# instructions</th> <th>Queues</th> <th>of Big O:</th>		Basic C# instructions	Queues	of Big O:
Understanding     • Stacks     • Space efficiency       Paper 1: Computer Systems     • Graphs     Completion of project proposal       Types of processor     Paper 1: Computer Systems     Paper 1: Computer Systems       Systems Software     Paper 2: Programming     Paper 2: Programming       Applications Generation     Boolean Algebra     Comparing Language       Software Development     Subtraction     Boolean Algebra       Types of Programming     Doclamation of the processor     Paper 2: Programming       Software Development     Subtraction     Boolean Algebra       Types of Programming     Data Types     Data Types       Big O notation of functions     Baper 2: Programming     Big O notation of functions       Sequence of delivery     (c) The eliferences between an abstraction and reality.     Arrays / Tuples / records     • Constant time       Studies     - Stacks     - Completion of functions     • Constant time       Sequence of delivery     (c) The eliferences between an abstraction and algebra     • Coucues     • Coucues       Stacks     - Constant time     • Constant time       Sequence of delivery     (c) The eliferences between an abstraction and reality.     • Arrays / Tuples / records     • Exponential       • Listy / Linked Lists     • Stacks     • Trees     Requirements of the NEA       • (c) The nature,		Sequence / Selection / Iteration	Lists / Linked Lists	Time efficiency
Understanding/ delivery     Paper 1: Computer Systems     Paper 1: Computer Systems     Paper 1: Computer Systems       Understanding / delivery     Paper 1: Computer Systems     Paper 1: Computer Systems     Paper 1: Computer Systems       Understanding / delivery     Structure and function of the processor Types of processor     Paper 1: Computer Systems     Paper 1: Computer Systems       Software Bevelopment Software Development Software Development Software Development     Paper 2: Programming Subroutines and recursion     Paper 2: Programming Subroutines and recursion     Paper 2: Programming Subroutines and recursion       Understanding / delivery     (a) The nature of abstraction.     (b) The need for abstraction (a) The nature of abstraction.     A arays / Tuples / records <ul> <li>(b) The need for abstraction,</li> <li>(c) The differences between an abstract</li> <li>(c) The differences between an abstract model for a variety of situations.</li> <li>(d) Identify the inputs and outputs for a given situation.</li> <li>(d) Identify the inputs and outputs for a given situation.</li> <li>(e) The nature, benefits and drawbacks of caching.</li> <li>(d) The neature, benefits and drawbacks of caching.</li> <li>(d) The neature, benefits and drawbacks of caching.</li> <li>(d) The neature, benefits and drawbacks of caching.</li> <li>(e) The neature, benefits and drawbacks of caching.</li> <li>(f) The neature of the reusable program components.</li> <li>(f) The neature of the reusable program components.</li> <li>(f) The neature of the reusable program components.</li> <li>(f) The neate for oreusable program components.</li> <li>(f) The neat</li></ul>			Stacks	Snace efficiency
Understanding / Bedivery     Paper 1: Computer Systems     Completion of project proposal Completion of Analysis       Understanding / Bedivery     Paper 1: Computer Systems     Paper 1: Computer Systems       Understanding / Bedivery     Paper 1: Computer Systems     Completion of Analysis       Understanding / Bedivery     Paper 1: Computer Systems     Computer Systems       Understanding / Bedivery     Paper 1: Computer Systems     Computer Systems       Completion of project proposal     Completion of Analysis       Understanding / Bedivery     Paper 1: Computer Systems     Computer Systems       Completion of project proposal     Completion of Analysis       Understanding / Bedivery     Paper 1: Computer Systems     Computer Systems       Completion of project proposal     Completion of Analysis     Paper 1: Computer Systems       Understanding / Bedivery     Structure and function.     Paper 2: Programming Abstraction.     Paper 2: Programming Subroutines and recursion     Paper 3: Computer Systems       I (D) The need for a bastract model for a variety of situation.     Stacks     Paper 3: Computer Systems     Paper 3: Completion of functions       (c) The differences between an abstract model for a variety of situation.     Stacks     Paper 3: Completion of aproject proposal Requirements of the NEA       (c) The nead for reusable program components.     Paper 3: Completion of aproject proposal Requirements of Analysis       (d) The neatu			Hash tables	NFA:
Understanding / Beguence of delivery         Image of processor Types of processor         Paper 1: Computer Systems Compressor, Encryption and Hashing Databases         Paper 1: Computer Systems Computing Systems Databases         Paper 1: Computer Systems Computing Systems Paper 2: Programming Subtraction         Paper 1: Computer Systems Computing Systems Databases         Paper 1: Computer Systems Computing Systems Paper 2: Programming Subtraction         Paper 1: Computer Systems Computing Systems Databases         Paper 1: Computer Systems Computing Systems Paper 2: Programming Subtraction         Paper 1: Computer Systems Computing Systems Boolean Algebra         Paper 2: Programming DataSystems Data Types of Programming Subtraction         Paper 2: Programming Subtraction         Paper 2: Programming Subtraction         Data Types for Computer Systems Data Types / records         Data Types / Records         NEA:           (a) The nature of abstraction (b) The need for abstraction (c) The differences between an abstraction and reality. <ul> <li>List / Linked Lists</li> <li>Graphs</li> <li>Tricking ahead</li> <li>Trees</li> <li>Trees</li> <li>Trees</li> <li>Trees</li> <li>Trees</li> <li>Trees</li> </ul>			Graphs	Completion of project proposal
Paper 1: Computer Systems     Paper 1: Computer Systems     Paper 1: Computer Systems       Structure and function of the processor     Compression, Encryption and Hashing     Databases       Types of processor     Databases     Networks       Systems Software     Boolean Algebra     Paper 2: Programming       Applications Generation     Software Development     Types of Programming       Types of Programming     Subroutines and recursion     Databases       Software Development     Subroutines and recursion     Data Types       Subraction     (a) The nature of abstraction.     (b) The ediferences between an abstraction and trains     • Arrays / Tuples / records       (b) The need for abstraction.     • Ustas / Luiss / Luisted Lists     • Logarithmic       (c) The differences between an abstraction and trains     • Stacks       (d) Devise an abstract model for a variety of studions.     • Trees       studion.     (b) Determine the proconditions for devising a solution to a problem.       (c) The nature, benefits and drawbacks of caching.     • Trees       (d) The nature of for enusable program components.     • Think procedurally       (d) The need for reusable program     • Trees			Trees	Completion of Analysis
Understanding/ sequence of delivery         Structure and function of the processor Types of processor Input, output and storage Systems Software Applications Generation Software Development Types of Programming Applications Generation.         Compression, Encryption and Hashing Databases         Computing related legislation           Understanding/ Sequence of delivery         Applications Generation Software Development Types of Programming Abstraction.         Boolean Algebra Boolean Al		Paper 1: Computer Systems	Paper 1: Computer Systems	Paper 1: Computer Systems
Understanding/     Types of processor input, output and storage Systems Software     Databases Networks     Moral and ethical issues       Videous Software     Databases input, output and storage Systems Software     Databases Networks     Moral and ethical issues       Applications Generation Software Development     Design of algorithms to solve a problem Subroutines and recursion     Design of algorithms to solve a problem Comparing algorithms       Software Development     Paper 2: Programming Abstraction     Boolean Algebra     Design of algorithms to solve a problem Comparing algorithms       Videous Paper 2: Programming Abstraction     In enture of abstraction.     Data Types     Data Types       (a) The neture of abstraction.     • Array / Tuples / records     • Constant time       (b) The need for abstraction.     • Array / Tuples / records     • Linear       (c) The differences between an abstraction and reality.     • Lists / Linked Lists     • Delynomial       (c) The inder of or abstract model for a variety of situations.     • Hash tables     Requirements of the NEA       (d) Identify the inputs and outputs for a given situation.     • Trees     Requirements of Analysis       (d) The need for reusable program components.     • Trees     Requirements of Analysis       (d) The need for reusable program components.     • Trink procedurally     • Hash tables     • Hash tables       (d) The need for reusable program components.     • Infinity procedurally		Structure and function of the processor	Compression, Encryption and Hashing	Computing related legislation
Understanding/ delivery         Input. output and storage Systems Software         Networks         Paper 2: Programming Analysis of problems         Paper 2: Programming Analysis of problems           Understanding/ delivery         Software Development         Boolean Algebra         Design of algorithms to solve a problem           Understanding/ delivery         Comparing algorithms         Software Development         Subroutines and recursion           Understanding/ delivery         (a) The nature of abstraction.         Data Structures <ul> <li>Constant time</li> <li>Exponential</li> <li>Usits / Linked Lists</li> <li>Exponential</li> <li>Exponential</li> <li>Stacks</li> <li>Graphs</li> <li>Graphs</li> <li>Requirements of a project proposal</li> <li>Requirements of a project proposal</li> <li>Requirements of Analysis</li> <li>Understanding/</li> <li>(d) The nature, benefits and drawbacks of caching.</li> <li>(d) The nearements.</li> <li>(d) the the means before the preconditions for devising a solution to a problem.</li> <li>(d) the the diversement be for enusable program</li> <li>(d) the the diversement be for enusable program</li> <li>(d) the the diversement be for enusable program</li></ul>		Types of processor	Databases	Moral and ethical Issues
Understanding / delivery         Systems Software Applications Generation Software Development         Web Technologies Boolean Algebra         Analysis of problems           Software Development         Boolean Algebra         Design of algorithms to solve a problem Comparing algorithms to solve a problem           Software Development         Boolean Algebra         Design of algorithms to solve a problem           Software Development         Subroutines and recursion         Functions           Types of Programming Language         Data Types         Big O notation of functions           Paper 2: Programming         Data Types         Data Types           Abstraction         - Arrays / Tuples / records         - Constant time           Abstraction.         - Arrays / Tuples / records         - Elsts / Linked Lists           Sequence of delivery         (c) The differences between an abstraction and ratitations.         - Elsts / Linked Lists         - Logarithmic           Thinking ahead         (a) Identify the inputs and outputs for a given situation.         - Trees         Requirements of the NEA Requirements of Analysis           (d) The nature, benefits and drawbacks of caching.         (d) The need for reusable program components.         - Trees         Requirements of Analysis		Input output and storage	Networks	Paper 2: Programming
Applications Generation       Boolean Algebra       Design of algorithms to solve a problem         Software Development       Paper 2: Programming       Comparing algorithms         Types of Programming       Data Types       Big O notation of functions         Paper 2: Programming       Data Types       Data Types         Abstraction       Data Types       Data Types         (a) The neart of abstraction.       Data Structures       Linear         (b) The need for abstraction and reality.       Queues       Exponential         (c) The differences between an abstract model for a variety of situations.       Graphs       NEA:         Thinking ahead       Graphs       Requirements of the NEA         (a) I dentify the inputs and outputs for a given situation.       (b) Determine the preconditions for devising a solution to a problem.       Trees         (c) The nature, benefits and drawbacks of caching.       (d) The need for reusable program       Trink procedurally		Systems Software	Web Technologies	Analysis of problems
Understanding /       Software Development       Paper 2: Programming       Comparing algorithms         Junderstanding /       Types of Programming Language       Subroutines and recursion       Big O notation of functions         Junderstanding /       (a) The nature of abstraction.       Data Structures       - Constant time         (b) The need for abstraction.       - Arrays / Tuples / records       - Polynomial         (d) The need for abstraction.       - Uitst / Linked Lists       - Dopontial         (d) Devise an abstract model for a variety of situations.       - Stacks       NEA:         Thinking ahead       - Trees       - Trees       Requirements of Analysis         (a) Identify the inputs and outputs for a given situation.       - Trees       NEA:       Requirements of Analysis         (b) Determine the preconditions for devising a solution to a problem.       - (c) The nature, benefits and drawbacks of caching.       - Trees       Requirements of Analysis         (d) The need for reusable program components.       - Think procedurally       - Liest - L		Applications Generation	Boolean Algebra	Design of algorithms to solve a problem
Understanding /       Subroutines and recursion       Functions         Understanding /       Subroutines and recursion       Big O notation of functions         (a) The nature of abstraction.       Data Types       Constant time         (b) The need for abstraction.       Queues       Lists / Linked Lists         (c) The differences between an abstract model for a variety of delivery       Understanding /       Stacks         (d) Devise an abstract model for a variety of situations.       Graphs       Requirements of the NEA         (a) Identify the inputs and outputs for a given situation.       Trees       Requirements of Analysis         (b) Determine the preconditions for devising a solution to a problem.       Trees       Requirements of Analysis         (d) The nature, benefits and drawbacks of caching.       (d) The nature, benefits and drawbacks of caching.       The nature, benefits and drawbacks of caching.       The nature benefits and benegram         (d) The need for reusable program       Think procedurally       Lists / Linked Lists       Lists / Linked Lists         (c) The differences between an abstract model for a variety of situation.       Trees       Requirements of a project proposal         (a) Identify the inputs and outputs for a given situation.       (b) Determine the preconditions for devising a solution to a problem.       Image: Components.       Image: Components.         (d) The need for reusable progra		Software Development	Paper 2: Programming	Comparing algorithms
Understanding / Sequece of delivery     Image: Paper 2: Programming Abstraction <ul> <li>(a) The nature of abstraction.</li> <li>(b) The need for abstraction.</li> <li>(c) The differences between an abstraction and reality.</li> <li>(d) Devise an abstract model for a variety of situations.</li> <li>(d) Devise an abstract model for a variety of situations.</li> <li>(d) Identify the inputs and outputs for a given situation.</li> <li>(b) Determine the preconditions for devising a solution to a problem.</li> <li>(c) The nature, benefits and drawbacks of caching.</li> <li>(d) The need for reusable program components.</li> <li>Think procedurally</li> </ul> <ul> <li>(b) Identify the represented for any backs of caching.</li> <li>(c) The nature, benefits and drawbacks of caching.</li> <li>(d) The need for reusable program components.</li> <li>Think procedurally</li> </ul> <ul> <li>(c) Identify the represented for evented for any backs of caching.</li> <li>(d) The need for reusable program</li> <li>(</li></ul>		Types of Programming Language	Subroutines and recursion	Functions
Understanding / Sequence of delivery       Constant time         (a) The nature of abstraction. (b) The need for abstraction. (c) The differences between an abstraction and reality.       Data Structures <ul> <li>Arrays / Tuples / records</li> <li>Queues</li> <li>Lists / Linked Lists</li> <li>Stacks</li> <li>Hash tables</li> <li>Graphs</li> <li>Itrees</li> </ul> <ul> <li>NEA:</li> <li>Requirements of the NEA</li> <li>Requirements of a project proposal</li> <li>Requirements of Analysis</li> </ul> (a) Identify the inputs and outputs for a given situation. <ul> <li>(b) Determine the preconditions for devising a solution to a problem.</li> <li>(c) The nature, benefits and drawbacks of caching.</li> <li>(d) The need for reusable program components.</li> <li>Think procedurally</li> <li>(c) Identify the inputs for a given situation.</li> <li>(b) Determine the preconditions for devising a solution to a problem.</li> <li>(c) The nature, benefits and drawbacks of caching.</li> <li>(d) The need for reusable program components.</li> <li>Think procedurally</li> </ul> <ul> <li>(b) bettermine the preconditions for devising a solution to a problem.</li> <li>(c) The nature, benefits and drawbacks of caching.</li> <li>(d) The need for reusable program components.</li> <li>Think procedurally</li> <li>(b) bettermine the preconditions for devising a solution to a problem.</li> <li>(c) Identify the need for reusable prog</li></ul>		Paper 2: Programming	Data Types	Big O notation of functions
Understanding / Sequence of delivery       (a) The nature of abstraction.       • Arrays / Tuples / records       • Linear         Understanding / Sequence of delivery       (c) The differences between an abstraction and reality.       • Arrays / Tuples / records       • Polynomial         (d) Devise an abstract model for a variety of situations.       • Lists / Linked Lists       • Logarithmic         Thinking ahead       • Graphs       • Requirements of the NEA         (a) Identify the inputs and outputs for a given situation.       • Trees       Requirements of Analysis         (b) Determine the preconditions for devising a solution to a problem.       • Trees       Requirements of Analysis         (d) The need for reusable program components.       • Think procedurally       • Integration       • Understanding         (b) Determine the preconditions for devising a solution to a problem.       • Outputs and outputs for a given situation to a problem.       • Trees       • Polynomial         (c) The nature, benefits and drawbacks of caching.       • Graphs       • Graphs       • Graphs       • Frees		Abstraction	Data Structures	Constant time
Understanding /       (b) The need for abstraction.       • Queues       • Polynomial         Understanding /       (c) The differences between an abstraction and reality.       • Queues       • Lists / Linked Lists         (d) Devise an abstract model for a variety of situations.       • Graphs       • Stacks       • NEA:         Thinking ahead       (a) Identify the inputs and outputs for a given situation.       • Trees       • Requirements of the NEA         (b) Determine the preconditions for devising a solution to a problem.       • (c) The nature, benefits and drawbacks of caching.       • The need for reusable program components.       • Think procedurally         (d) The need for reusable program       • (d) The need for reusable program		(a) The nature of abstraction.	Arrays / Tuples / records	Linear
Understanding / Sequence of delivery       (c) The differences between an abstraction and reality.       • Lists / Linked Lists       • Exponential Logarithmic         (d) Devise an abstract model for a variety of situations.       • Hash tables       • Requirements of the NEA         (a) Identify the inputs and outputs for a given situation.       • Trees       Requirements of Analysis         (b) Determine the preconditions for devising a solution to a problem.       • Trees       Requirements of Analysis         (c) The nature, benefits and drawbacks of caching.       (d) The need for reusable program components.       • Think procedurally         (c) The need for reusable program       • Think procedurally       • Think procedurally       • Think procedurally		(b) The need for abstraction.	Queues	Polynomial     Supercential
Sequence of delivery       reality.       • Stacks       NEA:         (d) Devise an abstract model for a variety of situations.       • Hash tables       Requirements of the NEA         (a) Identify the inputs and outputs for a given situation.       • Trees       Requirements of Analysis         (b) Determine the preconditions for devising a solution to a problem.       • Trees       Requirements of Analysis         (c) The nature, benefits and drawbacks of caching.       (d) The need for reusable program       • Think procedurally         (a) Think procedurally       • Trees       • Trees	Understanding /	(c) The differences between an abstraction and	Lists / Linked Lists	Exponential
delivery       (d) Devise an abstract model for a variety of situations.       • Hash tables       Requirements of the NEA         delivery       (d) Devise an abstract model for a variety of situations.       • Graphs       Requirements of a project proposal         Thinking ahead       • Trees       Requirements of Analysis         (a) Identify the inputs and outputs for a given situation.       • Determine the preconditions for devising a solution to a problem.       • Trees         (c) The nature, benefits and drawbacks of caching.       (d) The need for reusable program       • Hash tables       • Graphs         (d) Determine th.       • Trees       • Trees       • Trees	Sequence of	reality.	Stacks	• Logantininc
<ul> <li>situations.</li> <li>Graphs</li> <li>Graphs</li> <li>Thinking ahead</li> <li>Trees</li> <li>Requirements of a project proposal</li> <li>Requirements of Analysis</li> </ul>	delivery	(d) Devise an abstract model for a variety of	Hash tables	Requirements of the NEA
<ul> <li>Thinking ahead</li> <li>Trees</li> <li>Requirements of a project proposal</li> <li>Requirements of Analysis</li> <li>Requirements of Analysis</li> <li>Requirements of Analysis</li> <li>Requirements of Analysis</li> <li>C) The nature, benefits and drawbacks of caching.</li> <li>(d) The need for reusable program</li> <li>components.</li> <li>Think procedurally</li> <li>(a) Identify the need for reusable program</li> <li>(b) Identify the need for reusable program</li> <li>(c) The need for reusable program</li> <li>(c) The need for reusable program</li> <li>(d) The need for reusable program</li> <li>(e) Identify the need for reusable program</li> <li>(f) Identify the need for reusable program</li> <li>(g) Identify the need for reusable program</li> </ul>		situations.	Graphs	Requirements of a project proposal
<ul> <li>(a) Identify the inputs and outputs for a given situation.</li> <li>(b) Determine the preconditions for devising a solution to a problem.</li> <li>(c) The nature, benefits and drawbacks of caching.</li> <li>(d) The need for reusable program components.</li> <li>Think procedurally</li> <li>(a) Identify the sourcements of a methlem.</li> </ul>		Thinking ahead	Trees	Requirements of Analysis
situation. (b) Determine the preconditions for devising a solution to a problem. (c) The nature, benefits and drawbacks of caching. (d) The need for reusable program components. Think procedurally		(a) Identify the inputs and outputs for a given		Requirements of Analysis
<ul> <li>(b) Determine the preconditions for devising a solution to a problem.</li> <li>(c) The nature, benefits and drawbacks of caching.</li> <li>(d) The need for reusable program components.</li> <li>Think procedurally</li> <li>(a) Identify the segments of a problem.</li> </ul>		situation.		
solution to a problem. (c) The nature, benefits and drawbacks of caching. (d) The need for reusable program components. Think procedurally (a) Identify the segmentements of a methlem		(b) Determine the preconditions for devising a		
<ul> <li>(c) The nature, benefits and drawbacks of caching.</li> <li>(d) The need for reusable program components.</li> <li>Think procedurally</li> <li>(a) Identify the segments of a methage.</li> </ul>		solution to a problem.		
caching. (d) The need for reusable program components. Think procedurally (a) Identify the components of a method		(c) The nature, benefits and drawbacks of		
(d) The need for reusable program components. Think procedurally		caching.		
components. Think procedurally		(d) The need for reusable program		
Think procedurally		components.		
		Think procedurally		
(a) identify the components of a problem.		(a) Identify the components of a problem.		

	<ul> <li>(b) Identify the components of a solution to a problem.</li> <li>(c) Determine the order of the steps needed to solve a problem.</li> <li>(d) Identify sub-procedures necessary to solve a problem.</li> <li>Thinking logically</li> <li>(a) Identify the points in a solution where a decision has to be taken.</li> <li>(b) Determine the logical conditions that affect the outcome of a decision.</li> <li>(c) Determine how decisions affect flow through a program.</li> <li>Thinking concurrently</li> <li>(a) Determine the parts of a problem that can be tackled at the same time.</li> <li>(b) Outline the benefits and trade offs that</li> </ul>		
	(b) Outline the benefits and trade offs that might result from concurrent processing in a particular situation.		
Assessments	Knowledge check Programming Homework	PPE 1 Programming Homework	Year 12 PPE NEA Approved project proposal and analysis

Year 12 OCR Computer Science A-Level	<ul> <li>Curriculum Intent: The aims of this qualification are to enable learners to develop:</li> <li>An understanding and ability to apply the fundamental principles and concepts of computer science, including: abstraction, decomposition, logic, algorithms and data representation</li> <li>The ability to analyse problems in computational terms through practical experience of solving such problems, including writing programs to do so</li> <li>The capacity to think creatively, innovatively, analytically, logically, and critically</li> <li>The capacity to see relationships between different aspects of computer science</li> </ul>			
Year 12 Algorithms and programming Component 02	Term 1:	Term 2:	Term 3	
Topic Titles (in order of delivery)	Elements of computational thinking Problem solving and programming	Programming techniques Computational methods Algorithms	Algorithms	
Key knowledge / Retrieval topics	Understand what is meant by computational thinking How computers can be used to solve problems and programs can be written to solve them	The use of algorithms to describe problems and standard algorithms	The use of algorithms to describe problems and standard algorithms	
Understanding / Sequence of delivery	Thinking abstractly Thinking ahead Thinking procedurally Thinking logically Thinking concurrently	Algorithms (a) Analysis and design of algorithms for a given situation. (b) The suitability of different algorithms for a given task and data set, in terms of execution time and space. (c) Measures and methods to determine the efficiency of different algorithms, Big O notation (constant, linear, polynomial, exponential and logarithmic complexity).	Algorithms (d) Comparison of the complexity of algorithms. (e) Algorithms for the main data structures, (stacks, queues, trees, linked lists, depth-first (post-order) and breadth-first traversal of trees). (f) Standard algorithms (bubble sort, insertion sort, merge sort, quick sort, Dijkstra's shortest path algorithm, A* algorithm, binary search and linear search)	
Assessments				