


OCR's GCSE (9–1) in Computer Science will encourage students to:

- Understand and apply the fundamental principles and concepts of Computer Science, including abstraction, decomposition, logic, algorithms, and data representation.
- Analyse problems in computational terms through practical experience of solving such problems, including designing, writing and debugging programs.
- Think creatively, innovatively, analytically, logically and critically.
- Understand the components that make up digital systems, and how they communicate with one another and with other systems.
- Understand the impacts of digital technology to the individual and to wider society.
- Apply mathematical skills relevant to Computer Science.

		<h1>Dallam School</h1> <p>Curriculum overview</p>		<p><b>Department: Computer Science</b> <b>Year Group: Y10</b></p>	
AUTUMN		SPRING		SUMMER	
Half term 1	Half term 2	Half term 3	Half term 4	Half term 5	Half term 6
<b>Theme / Topic</b> <b>1.1 Systems Architecture</b>	<b>Theme / Topic</b> <b>1.2 Memory and storage</b>	<b>Theme / Topic</b> <b>1.3 Computers networks, connections and protocols</b>	<b>Theme / Topic</b> <b>1.4 Network security</b>	<b>Theme / Topic</b> <b>1.5 Systems software</b>	<b>Theme / Topic</b> <b>1.6 Ethical, legal, cultural and environmental impacts of digital technology</b>
<p>By the end of this half term pupils will know (<i>key knowledge, including tier 3 vocabulary</i>)</p>					
<p>What actions occur at each stage of the fetch-execute cycle. The role/purpose of each component and what it manages, stores, or controls during the fetch-execute cycle. The purpose of each register, what it stores (data or address). The difference between storing data and an address.</p> <p><b>Tier 3 vocab</b></p> <ul style="list-style-type: none"> <li>• ALU (Arithmetic Logic Unit) CU (Control Unit)</li> <li>• Cache</li> </ul>	<p>Why computers have primary storage            Why computers have secondary storage            Recognise a range of secondary storage devices/media            Why data must be stored in binary format            Familiarity with data units and moving between each            Calculate required storage capacity for a given set of files            Calculate file sizes of sound, images and text files            Ability to deal with binary numbers containing between 1 and 8 bits            How</p>	<p>The characteristics of LANs and WANs            Understanding of different factors that can affect the performance of a network, DNS's role in the conversion of a URL to an IP address            Concept of servers providing services            The Cloud: remote service provision (e.g. storage, software, processing)            Advantages and disadvantages of the Star and Mesh topologies            Compare benefits and drawbacks of wired versus wireless connection            IP</p>	<p>Threats posed to devices/systems, the purpose of the attack and methods to remove vulnerabilities.</p> <p><b>Tier 3 vocab</b></p> <ul style="list-style-type: none"> <li>• Malware</li> <li>• Social engineering,</li> <li>• Phishing</li> <li>• Denial of service attacks</li> <li>• SQL injection</li> <li>• Penetration testing</li> </ul>	<p>The purpose and functionality of operating systems and utility software.</p> <p><b>Tier 3 vocab</b></p> <ul style="list-style-type: none"> <li>• User interface</li> <li>• Memory management and multitasking</li> <li>• Peripheral management and drivers</li> <li>• User management</li> <li>• File management</li> </ul>	<p>Impacts of digital technology on wider society and Legislation relevant to Computer Science.</p> <p><b>Tier 3 vocab</b></p> <ul style="list-style-type: none"> <li>• Digital divide</li> <li>• E-waste</li> <li>• Open source</li> <li>• Proprietary</li> <li>• Artificial Intelligence</li> <li>• Machine Learning</li> </ul>

<ul style="list-style-type: none"> <li>Registers</li> <li>Von Neumann architecture MAR (Memory Address Register)</li> <li>MDR (Memory Data Register)</li> <li>Program Counter</li> <li>Accumulator</li> <li>Embedded systems</li> </ul>	<p>characters are represented in binary Each pixel has a specific colour, represented by a specific code Analogue sounds must be stored in binary Common scenarios where compression may be needed</p> <p><b>Tier 3 vocab</b></p> <ul style="list-style-type: none"> <li>RAM</li> <li>ROM</li> <li>Denary</li> <li>Hexadecimal</li> <li>Metadata</li> <li>Pixel</li> <li>Bit</li> <li>Nibble</li> <li>Byte</li> <li>Kilobyte</li> <li>Megabyte</li> <li>Gigabyte</li> <li>Terabyte</li> <li>Petabyte</li> </ul>	<p>addressing and the format of an IP address (IPv4 and IPv6) The principle of a (communication) protocol as a set of rules for transferring data How layers are used in protocols, and the benefits of using layers.</p> <p><b>Tier 3 vocab</b></p> <ul style="list-style-type: none"> <li>Client</li> <li>Server</li> <li>Peer to Peer</li> <li>Network Layers</li> <li>Network Protocols</li> <li>MAC address</li> <li>DNS</li> <li>Router</li> </ul>	<ul style="list-style-type: none"> <li>Anti-malware software</li> <li>Encryption</li> </ul>	<ul style="list-style-type: none"> <li>Encryption software</li> <li>Defragmentation</li> <li>Data compression</li> </ul>	
---	---	--	---	--	--

They will understand (*key concepts*)

<p>The role of the CPU, factors which affect performance and the role each register plays in the fetch decode execute cycle. The purpose and function of embedded systems.</p>	<p>How data is stored by a computer, why it is stored in a certain way primary and secondary storage is needed.</p>	<p>How data is transferred over a network and why this is useful. The impact network configuration can have on how data is sent and the need for agreed networking protocols.</p>	<p>The threats data on networks face, ways in which to these threats can be mitigated and the role software and training play in nullifying threats.</p>	<p>What each function of an operating system does, the features of a user interface and the purpose of utility software.</p>	<p>Technology introduces ethical, legal, cultural, environmental and privacy issues. How digital technology affects society. The purpose legislation and the specific actions it allows or prohibits in regards to technology.</p>
--	---	---	--	--	--

They will know how to (*key skills*)

<ul style="list-style-type: none"> <li>• I know a computer has no intelligence and will not perform any task unless programmed to do so.</li> <li>• I know a computer is a machine which is able to take in data via an input device, process data via the CPU and output data via an output device.</li> <li>• I can name the three tasks a CPU performs to process data.</li> <li>• I know the difference between hardware and software.</li> <li>• I know embedded systems are smaller systems within a larger system.</li> <li>• I know each core of a CPU processes data separately.</li> <li>• I know the more cores a CPU has the more instructions it can carry out simultaneously.</li> <li>• I can explain the role of cache memory.</li> <li>• I can list internal computer components.</li> </ul>	<ul style="list-style-type: none"> <li>• I know a computer has memory which acts as a temporary storage for programs &amp; data whilst a computer is switched on.</li> <li>• I know the difference between volatile &amp; non-volatile memory.</li> <li>• I know RAM is volatile.</li> <li>• I know ROM is non-volatile.</li> <li>• I recognise the difference between memory and storage.</li> <li>• I can look at a computer spec and determine which has the most memory.</li> <li>• I know the more RAM a Computer has the smoother processes will run.</li> <li>• I know ROM tells the computer how to boot up.</li> <li>• I know ROM is read only to the CPU but can be updated using a technique called flashing.</li> <li>• I know more RAM means that more applications can be open at once.</li> <li>• I know more RAM means more</li> </ul>	<ul style="list-style-type: none"> <li>• I know networking computers involves linking them together in order to share information.</li> <li>• I can list some advantages of networking computers.</li> <li>• I know some networks are wired and some are wireless.</li> <li>• I know what WAN &amp; LAN stand for.</li> <li>• I can recognise the following network topologies: bus, ring, star, partial-mesh &amp; full-mesh.</li> <li>• I know computers on a network must agree to use the same protocols in order to communicate.</li> <li>• I know networking comes with risks to data/system security.</li> <li>• I can define the difference between a LAN &amp; a WAN.</li> <li>• I know hardware is needed to create a network.</li> <li>• I know the role of a router in connecting networks.</li> <li>• I know the difference between</li> </ul>	<ul style="list-style-type: none"> <li>• I know computer systems are vulnerable to security threats.</li> <li>• I can list one or more security threat.</li> <li>• I can list good practice for setting hard to guess passwords.</li> <li>• I know data should be backed regularly to prevent loss.</li> <li>• I can define a brute force attack.</li> <li>• I can define malware and list several types.</li> <li>• I know the difference between a passive and an active threat.</li> <li>• I know what SQL injection is.</li> <li>• I know a firewall monitors network traffic and is designed to prevent security threats to a network.</li> <li>• I know it is the responsibility of the data holder to keep data secure and if this is not done they are in breach of the Data Protection Act 1998.</li> <li>• I can outline the difference between</li> </ul>	<p>Understand that:</p> <ul style="list-style-type: none"> <li>• Data is transferred between devices and the processor</li> <li>• This process needs to be managed</li> <li>• User management functions, e.g.: Allocation of an account § Access rights § Security, etc.</li> </ul> <p>File management, and the key features, e.g:</p> <ul style="list-style-type: none"> <li>• Naming</li> <li>• Allocating to folders Moving files §</li> <li>• Saving, etc.</li> </ul> <p>Understand that computers often come with utility software, and how this performs housekeeping tasks Purpose of the identified utility software and why it is required</p>	<ul style="list-style-type: none"> <li>• I know there is laws(legislation) which make certain activities on computer systems illegal.</li> <li>• I know people have a right for data stored about them to be private.</li> <li>• I can define cyberbullying &amp; give an example.</li> <li>• I am aware computer systems can have a negative effect on the environment.</li> <li>• I can define the term raw material.</li> <li>• I am aware not all countries &amp; people have the same access to digital devices &amp; services.</li> <li>• I can define the term cookie.</li> <li>• I know all devices have a lifespan.</li> <li>• I can list some of the negative effects computer systems may have on the environment.</li> <li>• I can identify a stakeholder in a given scenario.</li> <li>• I can explain how cookies are used &amp; how this may affect privacy.</li> </ul>
---	--	---	--	---	--

<ul style="list-style-type: none"> <li>• I can complete sections of a CPU diagram.</li> <li>• I can choose from true or false statements which describe CPU functions.</li> <li>• I can list the CPU registers.</li> <li>• I can recognise the difference between Von Neuman and Harvard architecture.</li> <li>• I can label all sections of a CPU.</li> <li>• I can list examples of embedded systems.</li> <li>• I can list difference types of computer system.</li> <li>• I can explain the purpose of all CPU registers.</li> <li>• I know the difference between operands and op codes.</li> <li>• I can describe the FDE cycle.</li> <li>• I can explain the need for embedded systems.</li> <li>• I can explain how the CPU and GPU work together to improve system performance.</li> <li>• I can explain the difference between L1, L2 &amp; L3 cache.</li> </ul>	<p>memory intensive programs can be run without slowing down the system.</p> <ul style="list-style-type: none"> <li>• I can list the processes the BIOS goes through to begin bootup.</li> <li>• I know what virtual memory is.</li> <li>• I know instructions are stored in RAM before the are fetched by the CPU.</li> <li>• I know the impact of increasing RAM capacity.</li> <li>• I can explain why virtual memory is needed.</li> <li>• I can explain the difference between static and dynamic RAM.</li> <li>• I can explain how the CPU implements virtual memory.</li> <li>• I can explain why virtual memory is not as fast as RAM.</li> <li>• I can order the different memory locations and the data access speed in relation to the CPU.</li> <li>• I can explain the need for ROM and RAM.</li> <li>• I can list impacts of using virtual memory on system performance.</li> </ul>	<p>a P2P and a client server network.</p> <ul style="list-style-type: none"> <li>• I can list some hardware needed to create a wireless network.</li> <li>• I can list some hardware needed to create a wired network.</li> <li>• I can identify factors which affect data transfer speed.</li> <li>• I can define bandwidth and latency.</li> <li>• I know the difference between a switch and a hub.</li> <li>• I can explain the difference between a MAC address &amp; an IP address.</li> <li>• I can order the stages of packet switching given a list.</li> <li>• I can explain how Wi-Fi uses radio waves to transmit data.</li> <li>• I can list the advantages &amp; disadvantages of a P2P and a client serve network.</li> <li>• I can explain how the following communication protocols work: POP, IMAP, HTTPS, FTP, SMTP.</li> </ul>	<p>full and incremental backups.</p> <ul style="list-style-type: none"> <li>• I know what social engineering is and I can give at least one example.</li> <li>• I list several Data Protection Act 1998 principles.</li> <li>• I can outline steps a network manager should take to keep their network safe from internal and external threats.</li> <li>• I explain what a DOS attack is and its impact on a system.</li> <li>• I can explain how using a VPN may make data being communicated over a public network less vulnerable.</li> <li>• I can list several malware threats, explain how they impact upon a security systems and how these threats can be mitigated against.</li> <li>• I can explain the role of antivirus software in preventing malware attacks, including the need to regularly update virus definitions.</li> <li>• I can look at a list of computer</li> </ul>		<ul style="list-style-type: none"> <li>• I can explain how computer misuse may negatively impact a stakeholder in a given scenario.</li> <li>• I know the difference between censorship and surveillance.</li> <li>• I can list some social media acceptable use policies.</li> <li>• I can explain how gaining raw materials may impact the environment.</li> <li>• I can explain why social media platforms have acceptable use policies &amp; give examples.</li> <li>• I can discuss a range of negative effects unequal access to digital services may cause.</li> <li>• I can explain factors which may determine device lifespan.</li> <li>• I can balance negative effects computer systems may have with benefits they may provide.</li> <li>• I can read a scenario &amp; identify several stakeholders &amp; issues.</li> </ul>
---	---	--	---	--	--

<ul style="list-style-type: none"> <li>• I can explain the difference between Harvard and Von Neuman architecture.</li> <li>• I know why embedded systems use Harvard architecture and can explain this.</li> <li>• I evaluate system uses and make recommendations as to best system for a given scenario.</li> <li>• I can fully explain the role, purpose and functions the CPU performs.</li> <li>• I can evaluate the impact clock speed, number of the cores, cache size, pipelining and super scalar architecture has on system performance.</li> <li>• I am able to fully discuss the impact system architecture has on system performance and can evaluate situations in which Von Neuman and Harvard architectures are most appropriate.</li> <li>• I can evaluate the implementation of RISC &amp; CISC.</li> </ul>	<ul style="list-style-type: none"> <li>• I can explain fully and with examples how RAM impacts upon system performance.</li> <li>• I can compare and contrast two or more computer specs and evaluate the impact of RAM will have on the respective systems whilst consistently applying my answer to the scenario context and using technical terminology accurately.</li> <li>• I am able to fully discuss the impact of varying types and specs of RAM, its capacity and the role the kernel plays in managing memory in terms of system performance whilst consistently applying my answer to the scenario context and using technical terminology accurately.</li> </ul>	<ul style="list-style-type: none"> <li>• I can list the four network layers &amp; explain what happens in each.</li> <li>• I can compare &amp; contrast wired &amp; wireless network advantages &amp; disadvantages.</li> <li>• I can fully explain packet switching including the information a packet is labelled with and the protocols involved.</li> <li>• I can list wired network mediums, there relative speeds &amp; characteristics.</li> <li>• I can explain what a VPN is and the advantages of using one.</li> <li>• I can explain Wi-Fi frequencies &amp; why different channels are needed.</li> <li>• I am able to fully evaluate the legal, usability, reliability, support and cost implications of an end users choice between network topologies including decisions regarding hardware to connect a LAN to a WAN using accurate SPAG and technical terms.</li> </ul>	<p>offenses and match them to the Computer Misuse Act 1990 breach.</p> <ul style="list-style-type: none"> <li>• I can define network forensics &amp; give an example.</li> <li>• I can explain, with examples the role encryption plays in keeping data secure.</li> <li>• I can explain the purpose of penetration testing and the role it plays in keeping networks secure.</li> <li>• I can explain how a firewall scrutinises data packets to determine whether they can pass through a network &amp; prevent threats.</li> <li>• I can explain SQL injection fully, using examples of SQL statements and the data criminals may want to access and the role input sanitisation plays in preventing this.</li> <li>• I am able to fully explain threats a computer system may face (both passive and active) and how these can be mitigated against; using accurate</li> </ul>		<ul style="list-style-type: none"> <li>• I know the following pieces of legislation &amp; can give examples of offences which breach these pieces of legislation: The Data Protections Act 1998, The Computer Misuse Act 1990, The Copyrights Designs &amp; Patents Act 1988, Creative commons licence &amp; The freedom of Information Act 200.</li> <li>• I can discuss how all of the acts above would impact upon different stakeholders if breached.</li> <li>• I understand some cultures have different views to my own on morality &amp; ethics, &amp; how this may shape their views on digital services.</li> <li>• I am able to fully discuss and evaluate: ethical, legal, environmental, cultural &amp; moral issues from a wide range of contexts; with specific examples which support my arguments from</li> </ul>
--	---	---	--	--	--

		<ul style="list-style-type: none"><li>I am able to fully discuss and evaluate the ethical, legal, cultural and environmental issues surrounding networking such as: the dark web, unequal access, freedom of speech &amp; privacy from the point of view of several stakeholders whilst consistently applying my answer to the scenario context and using technical terminology accurately.</li></ul>	<p>SPAG and technical terms.</p> <ul style="list-style-type: none"><li>I am able to fully discuss and evaluate the ethical, legal &amp; moral issues surrounding system security such as: hacking, malware, &amp; adequate network policy from the point of view of several stakeholders whilst consistently applying my answer to the scenario context and using technical terminology accurately.</li></ul>		<p>the point of view of several stakeholders, whilst consistently applying my answer to the scenario context and using technical terminology accurately.</p>
--	--	---	---	--	--



# Dallam School

Curriculum overview

Department: Computer Science  
Year Group: Y11

AUTUMN		SPRING		SUMMER	
Half term 1	Half term 2	Half term 3	Half term 4	Half term 5	
<b>Theme / Topic</b> 2.1 Algorithms	<b>Theme / Topic</b> 2.2 Programming fundamentals	<b>Theme / Topic</b> 2.3 Producing robust programs	<b>Theme / Topic</b> 2.4 Boolean logic	<b>Theme / Topic</b> 2.5 Programming languages and Integrated Development Environments	

By the end of this half term pupils will know (*key knowledge, including tier 3 vocabulary*)

<p>Principles of computational thinking, algorithm design and techniques including flowcharts and pseudocode. Key searching and sorting algorithms.</p> <p><b>Tier 3 vocab</b></p> <ul style="list-style-type: none"> <li>• Abstraction</li> <li>• Decomposition</li> <li>• Algorithmic thinking</li> <li>• Pseudocode</li> <li>• Flowcharts</li> <li>• Reference language/high-level programming language</li> <li>• Binary search</li> <li>• Linear search</li> <li>• Bubble sort</li> <li>• Merge sort</li> <li>• Insertion sort</li> </ul>	<p>The use of the basic programming constructs used to control the flow of a program, common arithmetic and Boolean operators, data types and basic data structures. Basic string manipulation, file handling and use of SQL to query tables.</p> <p><b>Tier 3 vocab</b></p> <ul style="list-style-type: none"> <li>• Variables</li> <li>• Constants</li> <li>• Operators</li> <li>• Sequence</li> <li>• Selection</li> <li>• Iteration (count- and condition-controlled loops)</li> <li>• Integer</li> <li>• Real</li> <li>• Boolean</li> <li>• Char</li> <li>• string</li> <li>• Casting</li> <li>• Modulo division</li> <li>• Integer Division</li> </ul>	<p>The need for and techniques associated with defensive design considerations &amp; maintainability when coding in a high-level language. The purpose of testing in refining algorithms.</p> <p><b>Tier 3 vocab</b></p> <ul style="list-style-type: none"> <li>• Authentication</li> <li>• Indentation</li> <li>• Iterative</li> <li>• Terminal</li> <li>• Normal</li> <li>• Boundary</li> <li>• Invalid</li> <li>• Erroneous</li> </ul>	<p>How to draw simple logic diagrams using the operators AND, OR and NOT along with the corresponding truth table.</p> <p><b>Tier 3 vocab</b></p> <ul style="list-style-type: none"> <li>• Conjunction</li> <li>• Disjunction</li> <li>• Negation</li> <li>• Logic diagram</li> <li>• Truth table</li> </ul>	<p>Characteristics and purpose of different levels of programming language. The purpose of translators and characteristics of a compiler and an interpreter. Common tools and facilities available in an Integrated Development Environment (IDE).</p> <p><b>Tier 3 vocab</b></p> <ul style="list-style-type: none"> <li>• Editors</li> <li>• Error diagnostics</li> <li>• Run-time environment</li> <li>• Translators</li> <li>• High level language</li> <li>• Low level language</li> </ul>	
--	--	---	--	--	--

	<ul style="list-style-type: none"> <li>• Concatenation</li> <li>• Slicing</li> <li>• Procedure</li> <li>• Function</li> <li>• Argument</li> <li>• Parameter</li> </ul>				
They will understand ( <i>key concepts</i> )					
Understanding the practical application of computational thinking in problem solving as well as understanding some of the key algorithms used in Computer Science.	Practical use of the data types, subprograms, program structures, string manipulation, data structures, file handling in a high-level language within the classroom. Ability to write SQL statements to manipulate a data set.	Understanding of the issues a programmer should consider to ensure that a program caters for all likely input values. Understand how to deal with invalid data in a program, the difference between testing modules of a program during development and testing the program at the end of production. The ability to test using a range of data. The ability to identify and correct logic and syntax errors.	Understanding of how to create, complete or edit logic diagrams and truth tables for given scenarios. Ability to work with more than one gate in a logic diagram.	The differences between high and low-level programming languages. The need for translators and differences, benefits and draw backs of using a compiler or an interpreter.	
They will know how to ( <i>key skills</i> )					
<p>I know an algorithm is a set of steps used to carry out a task or solve a problem.</p> <p>I know algorithms can be represented as a flow chart or pseudocode.</p> <p>I know there are several algorithms that may solve the same problem.</p> <p>I know some algorithms are more efficient than others.</p> <p>I can define abstraction, decomposition &amp; algorithmic thinking.</p>	<ul style="list-style-type: none"> <li>• I can solve small parts of a problem written in sequence.</li> <li>• I can assign values to variables.</li> <li>• I use sensible variable names &amp; use a consistent naming convention.</li> <li>• I can make use of selection within my pseudocode to control program flow.</li> <li>• I understand when to use variables and constants</li> </ul>	<ul style="list-style-type: none"> <li>• I know it is important to test programs fully before release.</li> <li>• I know programs may not always do what a programmer intended.</li> <li>• I know errors in programs are called bugs.</li> <li>• I know the purpose of testing is to find &amp; then fix bugs.</li> <li>• I can give examples of what may cause program bugs.</li> </ul>	<ul style="list-style-type: none"> <li>• I can combine Boolean operators using AND, OR and NOT.</li> <li>• I can apply logical operators in truth tables to solve problems.</li> <li>• I can recognise of each gate symbol.</li> <li>• I can use valid notation to create a Boolean equation of a logic circuit.</li> </ul>	<ul style="list-style-type: none"> <li>• I know the difference between high and low-level programming languages.</li> <li>• I can give benefits and draw backs of high and low-level programming languages.</li> <li>• I can explain why a translator is needed.</li> <li>• I can state the difference between a compiler and an interpreter.</li> <li>• I can give benefits and draw backs of</li> </ul>	



<p>I can complete an unfinished flow chart to represent a solution.</p> <p>I know the difference between “brute force” &amp; “divide &amp; conquer”.</p> <p>I can name two search algorithms.</p> <p>I can name three sorting algorithms.</p> <p>I can use a flow chart to represent partial solutions to a problem.</p> <p>I can use basic pseudocode statements to solve parts of a problem.</p> <p>I can identify where bugs in an algorithms may result in an error.</p> <p>I can explain the difference between iterative &amp; terminal testing.</p> <p>I can use decomposition to break problems down into smaller parts &amp; make them easier to solve.</p> <p>I can use abstraction to remove unnecessary detail from a problem and make it easier to solve.</p> <p>I can use algorithmic thinking to come up with a set of steps to solve a problem.</p> <p>I can use all of the correct symbols of a flow chart to represent the solution to a problem.</p>	<p>within my pseudocode.</p> <ul style="list-style-type: none"> <li>• I can make use of iteration to improve the efficiency of my solutions.</li> <li>• I know to use indentation after conditional and iterative statements.</li> <li>• I can make use of simple inbuilt functions such as print &amp; input.</li> <li>• I can apply simple arithmetic functions such as: + / * -.</li> <li>• I can increment and decrement variable values within my pseudocode.</li> <li>• I can select the most appropriate data type from Boolean, float/real, integer &amp; string.</li> <li>• I can make use of simple logical operators such as &gt;, &lt; &amp; == to write selective statements.</li> <li>• I can output variable values along side strings using: ,.</li> <li>• I can join (concatenate) strings using the +.</li> <li>• I can define functions to make my pseudocode more efficient.</li> </ul>	<ul style="list-style-type: none"> <li>• I know some bugs occur as the user enters data the program is not designed for.</li> <li>• I know what defensive design is.</li> <li>• I can explain why a programmer may comment their code as they develop it.</li> <li>• I can explain why indentation is used in programming.</li> <li>• I know what program misuse is.</li> <li>• I can apply/input test data to a program to accurately test the outcome.</li> <li>• I can explain how validation is used to prevent unexpected input.</li> <li>• I can list several types of validation check and give examples of how they would be used.</li> <li>• I can give examples of how a programmer may design their program defensively.</li> <li>• I can explain the difference between validation and verification.</li> <li>• I can explain the need to test</li> </ul>		<p>compilers and an interpreter.</p> <ul style="list-style-type: none"> <li>• I can state tools that an IDE provides. I can explain how each of the tools and facilities an IDE provides can be used to help a programmer develop a program.</li> <li>• I have practical experience of using a range of tools within an IDE.</li> </ul>	
---	---	---	--	---	--

<p>I know the difference between a syntax &amp; a logic error.</p> <p>I know which search algorithms to perform on an ordered or unordered.</p> <p>I can identify where errors in an algorithm may occur using test data.</p> <p>I can suggest improvements to an algorithm to fix errors(bugs) or make it more efficient.</p> <p>I know can give examples of the following types of test data in a given scenario: erroneous, boundary &amp; normal.</p> <p>I know the advantages and disadvantages of both searching algorithms.</p> <p>I can demonstrate how all three sorting algorithms work.</p> <p>I can demonstrate how both searching algorithms work.</p> <p>I know the advantages &amp; disadvantages of all three sorting algorithms.</p> <p>I can use correct pseudocode to fully represent all parts of a problem solution.</p> <p>I can explain which sorting algorithms is most efficient for specific data sets and why.</p>	<ul style="list-style-type: none"> <li>• I can call functions making use of parameter passing.</li> <li>• I can use the Boolean operators and, or &amp; not within selective or iterative constructs to control program flow.</li> <li>• I can use arrays to store multiple values under one heading refer to items within the array using index.</li> <li>• I can use the following logical operators: !=, &gt;= and &lt;= .</li> <li>• I can use the file handling techniques open and read.</li> <li>• I can use the file handling techniques close and write.</li> <li>• I can use casting functions such as str(), int() and flt() to change variable data types.</li> <li>• I can introduce validation to code to ensure data being input is sensible.</li> <li>• I can select and implement the most appropriate iterative techniques for a scenario from: For &amp; While loops.</li> </ul>	<p>programs with a variety of test data.</p> <ul style="list-style-type: none"> <li>• I know what erroneous, extreme &amp; normal tests are and could give examples for a given program.</li> <li>• I can select suitable test data for a program.</li> <li>• I know programs are rarely ever finished &amp; the programming team will update and fix errors throughout a programs life span.</li> <li>• I can identify bugs in a program &amp; fix them.</li> <li>• I can list</li> <li>• I can explain, with examples why input sanitisation prevents bugs or program misuse.</li> <li>• I know the difference between syntax and logic errors.</li> <li>• I can explain the difference between final and iterative testing.</li> <li>• I can define four test measures programs will undergo before release.</li> <li>• I can find syntax errors in code and explain how they can be corrected.</li> <li>• I can explain the advantages and</li> </ul>			
---	---	---	--	--	--

<p>I can explain which searching algorithms is efficient for specific data sets &amp; why.</p> <p>I can view a data set and apply the most efficient searching or sorting algorithms showing all steps of my working.</p> <p>I can calculate best &amp; worst case scenario comparisons.</p> <p>I am able to evaluate algorithm efficiency using Big O notation.</p>	<ul style="list-style-type: none"> <li>• I can use the file handling techniques append to edit data in files.</li> <li>• I can make use of more complex arithmetic operators within pseudocode such as mod %, div // and ** exponent.</li> <li>• I can write count controlled loops to repeat steps a set amount of times.</li> <li>• I can make use of conditional iteration to select, view and append records in an external data files.</li> <li>• I can uses nested iteration to perform tasks within pseudocode.</li> <li>• I can make use of complex inbuilt functions such as len(), random() and upper() to make solutions more efficient.</li> <li>• I can solve all parts of a problem &amp; use all the techniques listed above appropriately in all cases.</li> <li>• I can create pseudocode which is commented fully and explain what all sections do &amp; why techniques have been used so</li> </ul>	<p>disadvantages of final and iterative testing.</p> <ul style="list-style-type: none"> <li>• I can explain the need for program patches.</li> <li>• I can generate suitable test data and apply it to a program.</li> <li>• I can fully explain the software development life cycle.</li> <li>• I can explain how user input may be used to exploit weaknesses in a program &amp; how this can be prevented using validation, verification &amp; input sanitisation.</li> <li>• I can design and implement a test plan which would ensure a robust and fully functional program.</li> <li>• I can explain the need for program maintainability &amp; evaluate factors which impact on this.</li> <li>• I am able to fully discuss and evaluate the need for &amp; practical methods which will produce robust programs; using specific examples and considering the client, end-user</li> </ul>			
--	--	--	--	--	--

	<p>that someone else is able to fully understand my solution.</p> <ul style="list-style-type: none"><li>• I can create solutions which are fully efficient making use of recursion, parameter passing, validation &amp; input sanitisation.</li></ul>	<p>and programmer, whilst consistently applying my answer to the scenario context and using technical terminology accurately.</p>			
--	---	---	--	--	--