COMPUTER SCIENCE 8520 PLC

Student Name: _____ Tutor Grp: _____

Please use the "Personal Learning Checklist" below to rate your knowledge and understanding of the AQA GCSE Computer Science 8520 course. Further information about each element of the "Subject Content" can be found at:

http://www.aqa.org.uk/subjects/computer-science-and-it/gcse/computer-science-8520/subject-content

		Knowledge & Understanding Rating		
Unit	**Subject Content**	**Insecure**	**Intermediate**	**Secure**
3.1 Fundamentals of algorithms	**3.1.1 Representing algorithms**			
	Understand and explain the term 'algorithm'			
	Understand and explain the term 'decomposition'			
	Understand and explain the term 'abstraction'			
	Use a systematic approach to problem solving and algorithm creation representing those algorithms using pseudo-code and flowcharts			
	Explain simple algorithms in terms of their inputs, processing and outputs			
	Determine the purpose of simple algorithms			
	3.1.2 Efficiency of algorithms			
	Understand that more than one algorithm can be used to solve the same problem			
	Compare the efficiency of algorithms explaining how some algorithms are more efficient than others in solving the same problem			
	3.1.3 Searching algorithms			
	Understand and explain how the linear search algorithm works			
	Understand and explain how the binary search algorithm works			
	Compare and contrast linear and binary search algorithms			
	3.1.4 Sorting algorithms			
	Understand and explain how the merge sort algorithm works			
	Understand and explain how the bubble sort algorithm works			
	Compare and contrast merge sort and bubble sort algorithms			
3.2 Programming	**3.2.1 Data types**			
	Understand the concept of a data type			
	Understand and use the following appropriately: • integer • real • Boolean • character • string			
	3.2.2 Programming concepts			

		Knowledge & Understanding Rating		
Unit	**Subject Content**	**Insecure**	**Intermediate**	**Secure**
	Use, understand and know how the following statement types can be combined in programs: • variable declaration • constant declaration • assignment • iteration • selection • subroutine (procedure/function)			
	Use definite and indefinite iteration, including indefinite iteration with the condition(s) at the start or the end of the iterative structure			
	Use nested selection and nested iteration structures			
	Use meaningful identifier names and know why it is important to use them			
	3.2.3 Arithmetic operations in a programming language			
	Be familiar with and be able to use: • addition • subtraction • multiplication • real division • integer division, including remainders			
	3.2.4 Relational operations in a programming language			
	Be familiar with and be able to use: • equal to • not equal to • less than • greater than • less than or equal to • greater than or equal to			
	3.2.5 Boolean operations in a programming language			
	Be familiar with and be able to use: • NOT • AND • OR			
	3.2.6 Data structures			
	Understand the concept of data structures			
	Use arrays (or equivalent) in the design of solutions to simple problems			
	Use records (or equivalent) in the design of solutions to simple problems			
	3.2.7 Input/output and file handling			
	Be able to obtain user input from the keyboard			
	Be able to output data and information from a program to the computer display			
	Be able to read/write from/to a text file			
	3.2.8 String handling operations in a programming language			
	Understand and be able to use: • length • position			

		Knowledge & Understanding Rating		
Unit	**Subject Content**	**Insecure**	**Intermediate**	**Secure**
	• substring • concatenation • convert character to character code • convert character code to character • string conversion operations			
	3.2.9 Random number generation in a programming language			
	Be able to use random number generation			
	3.2.10 Subroutines (procedures and functions)			
	Understand the concept of subroutines			
	Explain the advantages of using subroutines in programs			
	Describe the use of parameters to pass data within programs			
	Use subroutines that return values to the calling routine			
	Know that subroutines may declare their own variables, called local variables, and that local variables usually: • only exist while the subroutine is executing • are only accessible within the subroutine			
	Use local variables and explain why it is good practice to do so			
	3.2.11 Structured programming			
	Describe the structured approach to programming			
	Explain the advantages of the structured approach			
	3.2.12 Robust and secure programming			
	Be able to write simple data validation routines			
	Be able to write simple authentication routines			
	Be able to select suitable test data that covers normal (typical), boundary (extreme) and erroneous data			
	Be able to justify the choice of test data			
	3.2.13 Classification of programming languages			
	Know that there are different levels of programming language: • low-level language • high-level language			
	Explain the main differences between low-level and high-level languages			
	Know that machine code and assembly language are considered to be low-level languages and explain the differences between them			
	Understand that ultimately all programming code written in high-level or assembly languages must be translated into machine code			
	Understand that machine code is expressed in binary and is specific to a processor or family of processors			
	Understand the advantages and disadvantages of low-level language programming compared with high-level language programming			

		Knowledge & Understanding Rating		
Unit	**Subject Content**	**Insecure**	**Intermediate**	**Secure**
	Understand that there are three common types of program translator: • interpreter • compiler • assembler			
	Explain the main differences between these three types of translator			
	Understand when it would be appropriate to use each type of translator			
3.3 Fundamentals of data representation	**3.3.1 Number bases**			
	Understand the following number bases: • decimal (base 10) • binary (base 2) • hexadecimal (base 16)			
	Understand that computers use binary to represent all data and instructions			
	Explain why hexadecimal is often used in computer science			
	3.3.2 Converting between number bases			
	Understand how binary can be used to represent whole numbers			
	Understand how hexadecimal can be used to represent whole numbers			
	Be able to convert in both directions between: • binary and decimal • binary and hexadecimal • decimal and hexadecimal			
	3.3.3 Units of information			
	Know that: • a bit is the fundamental unit of information • a byte is a group of 8 bits			
	Know that quantities of bytes can be described using prefixes			
	Know the names, symbols and corresponding values for the decimal prefixes: • kilo, 1 kB is 1,000 bytes • mega, 1 MB is 1,000 kilobytes • giga, 1 GB is 1,000 Megabytes • tera, 1 TB is 1,000 Gigabytes			
	3.3.4 Binary arithmetic			
	Be able to add together up to three binary numbers			
	Be able to apply a binary shift to a binary number			
	Describe situations where binary shifts can be used			
	3.3.5 Character encoding			
	Understand what a character set is and be able to describe the following character encoding methods: • 7-bit ASCII • Unicode			
	Understand that character codes are commonly grouped and run in sequence within encoding tables			

		Knowledge & Understanding Rating		
Unit	**Subject Content**	**Insecure**	**Intermediate**	**Secure**
	Describe the purpose of Unicode and the advantages of Unicode over ASCII			
	Know that Unicode uses the same codes as ASCII up to 127			
	3.3.6 Representing images			
	Understand what a pixel is and be able to describe how pixels relate to an image and the way images are displayed.			
	Describe the following for bitmaps: • size in pixels • colour depth			
	Describe how a bitmap represents an image using pixels and colour depth			
	Describe using examples how the number of pixels and colour depth can affect the file size of a bitmap image			
	Calculate bitmap image file sizes based on the number of pixels and colour depth			
	Convert binary data into a black and white image			
	Convert a black and white image into binary data			
	3.3.7 Representing sound			
	Understand that sound is analogue and that it must be converted to a digital form for storage and processing in a computer			
	Understand that sound waves are sampled to create the digital version of sound			
	Describe the digital representation of sound in terms of: • sampling rate • sample resolution			
	Calculate sound file sizes based on the sampling rate and the sample resolution			
	3.3.8 Data compression			
	Explain what data compression is			
	Understand why data may be compressed and that there are different ways to compress data			
	Explain how data can be compressed using Huffman coding			
	Be able to interpret Huffman trees			
	Be able to calculate the number of bits required to store a piece of data compressed using Huffman coding			
	Be able to calculate the number of bits required to store a piece of uncompressed data in ASCII			
	Explain how data can be compressed using run length encoding (RLE)			
	Represent data in RLE frequency/data pairs			

		Knowledge & Understanding Rating		
Unit	Subject Content	Insecure	Intermediate	Secure
3.4 Computer systems	**3.4.1 Hardware and software**			
	Define the terms 'hardware' and 'software' and understand the relationship between them			
	3.4.2 Boolean logic			
	Construct truth tables for the following logic gates: • NOT • AND • OR			
	Construct truth tables for simple logic circuits			
	Interpret the results of simple truth tables			
	Create, modify and interpret simple logic circuit diagrams			
	Explain what is meant by: • system software • application software			
	Give examples of both types of software			
	Understand the need for, and functions of, operating systems (OS) and utility programs			
	Understand that the OS handles management of the: • processor(s) • memory • I/O devices • applications • security			
	3.4.4 Systems architecture			
	Explain the 'Von Neumann architecture'			
	Explain the role and operation of main memory and the following major components of a central processing unit (CPU): • arithmetic logic unit • control unit • clock • bus			
	Explain the effect of the following on the performance of the CPU: • clock speed • number of processor cores • cache size • cache type			
	Understand and explain the 'Fetch-Execute cycle'			
	Understand the differences between main memory and secondary storage			
	Understand the differences between RAM and ROM			
	Understand why secondary storage is required			
	Be aware of different types of secondary storage (solid state, optical and magnetic)			
	Explain the operation of solid state, optical and magnetic storage			
	Discuss the advantages and disadvantages of solid state, optical and magnetic storage			

		Knowledge & Understanding Rating		
Unit	**Subject Content**	**Insecure**	**Intermediate**	**Secure**
	Explain the term 'cloud storage'.			
	Explain the advantages and disadvantages of cloud storage when compared to local storage			
	Understand the term 'embedded system' and explain how an embedded system differs from a non-embedded system			
3.5 Fundamentals of computer networks	**3.5 Fundamentals of computer networks**			
	Define what a computer network is			
	Discuss the benefits and risks of computer networks			
	Describe the main types of computer network including: • Personal Area Network (PAN) • Local Area Network (LAN) • Wide Area Network (WAN)			
	Understand that networks can be wired or wireless			
	Discuss the benefits and risks of wireless networks as opposed to wired networks			
	Explain the following common network topologies: • star • bus			
	Define the term ‘network protocol’			

	Explain the purpose and use of common network protocols including:
	• Ethernet
	• Wi-Fi
	• TCP (Transmission Control Protocol)
	• UDP (User Datagram Protocol)
	• IP (Internet Protocol)
	• HTTP (Hypertext Transfer Protocol)
	• HTTPS (Hypertext Transfer Protocol Secure)
	• FTP (File Transfer Protocol)
	• Email protocols:
	  ○ SMTP (Simple Mail Transfer Protocol)
	  ○ IMAP (Internet Message Access Protocol)			

	Understand the need for, and importance of, network security			
	Explain the following methods of network security: • authentication • encryption • firewall • MAC address filtering			
	Describe the 4 layer TCP/IP model: • application layer • transport layer • network layer • data link layer			
	Understand that the HTTP, HTTPS, SMTP, IMAP and FTP protocols operate at the application layer			
	Understand that the TCP and UDP protocols operate at the transport layer			
	Understand that the IP protocol operates at the network layer			

		Knowledge & Understanding Rating		
Unit	Subject Content	Insecure	Intermediate	Secure
3.6 Fundamentals of cyber security	**3.6 Fundamentals of cyber security**			
	Be able to define the term 'cyber security' and be able to describe the main purposes of cyber security			
	3.6.1 Cyber security threats			
	Understand and be able to explain the following cyber security threats: • social engineering techniques • malicious code • weak and default passwords • misconfigured access rights • removable media • unpatched and/or outdated software			
	Explain what penetration testing is and what it is used for			
	3.6.1.1 Social engineering			
	Define the term social engineering			
	Describe what social engineering is and how it can be protected against			
	Explain the following forms of social engineering: • blagging (pretexting) • phishing • pharming • shouldering (or shoulder surfing)			
	3.6.1.2 Malicious code			
	Define the term 'malware'			
	Describe what malware is and how it can be protected against.			
	Describe the following forms of malware: • computer virus • trojan • spyware • adware			
	3.6.2 Methods to detect and prevent cyber security threats			
	Understand and be able to explain the following security measures: • biometric measures (particularly for mobile devices) • password systems • CAPTCHA (or similar) • using email confirmations to confirm a user's identity • automatic software updates			
3.7	**3.7 Ethical, legal and environmental impacts of digital technology on wider society, including issues of privacy**			
	Explain the current ethical, legal and environmental impacts and risks of the following digital technology on society: • cyber security • mobile technologies • wireless networking • cloud storage • theft of computer code			

		Knowledge & Understanding Rating		
Unit	**Subject Content**	**Insecure**	**Intermediate**	**Secure**
	• issues around copyright of algorithms • cracking • hacking • wearable technologies • computer based implants			