PENSBY HIGH SCHOOL

EXAMINATION GUIDE 2021-23

Subject	Computer Science
Course code	Computer Science (9-1) - J277
Website address	https://www.ocr.org.uk/qualifications/gcse/computer-science-j277-from-2020/
Provisional examination dates	Paper 1: J277/01 Computer Systems 50%: **TBC** Paper 2: J277/ 02 Computational thinking, algorithms and programming 50%: **TBC**
GCSE grade type awarded	9-1
Programming Project	20 hour programming booklets to be completed – Feeds into Paper 2 questions.
Revision books	***Clear Revise:*** *https://www.pgonline.co.uk/resources/computer-science/gcse-ocr/clearrevise-j277/* *Revision booklets printed and tagged to exercise books.*
Useful websites	• https://student.craigndave.org/ • GCSE OCR J277 Computer Science Videos – Craig 'n' Dave | Students (craigndave.org) • https://quizlet.com/en-gb/content/ocr-gcse-computer-science-flashcards • https://smartrevise.online/ • https://senecalearning.com/ • https://papacomputerscience.wordpress.com/gcse-j277-new-overview/ • https://youtu.be/uMDQiZg8rWE • https://papacomputerscience.files.wordpress.com/2020/07/cs-gcse-glossary.pdf • https://app.memrise.com/course/403879/ocr-gcse-computing-keywords/ • https://www.bbc.co.uk/bitesize/examspecs/zmtchbk • https://www.canyoucompute.co.uk/intro-lessons.html • https://www.ocr.org.uk/students/exam-support/revision/ • https://www.w3schools.com/python/default.asp

1.1 Systems Architecture

Confidence	Clarification	✓
1.1.1 Architecture of the CPU		
☹ 😐 ☺	I can explain the **purpose of the CPU**	
☹ 😐 ☺	I can explain what actions occur at each stage of the **fetch-execute cycle**	
☹ 😐 ☺	I can explain the role/purpose of the **(ALU) Arithmetic Logic Unit** and what data it stores during the fetch-execute cycle	
☹ 😐 ☺	I can explain the role/purpose of the **(CU) Control Unit** and what it manages and controls during the fetch-execute cycle	
☹ 😐 ☺	I can explain the role/purpose of the **Cache** and what data it stores during the fetch-execute cycle	
☹ 😐 ☺	I can explain the role/purpose of the **(MAR) Memory Address Register** in the Von Neumann architecture and what addresses it stores during the fetch-execute cycle	
☹ 😐 ☺	I can explain the role/purpose of the **(MDR) Memory Data Register** in the Von Neumann architecture and what data it stores during the fetch-execute cycle	
☹ 😐 ☺	I can explain the role/purpose of the **(PC) Program Counter** in the Von Neumann architecture and what data/address it stores during the fetch-execute cycle	
☹ 😐 ☺	I can explain the role/purpose of the **Accumulator** in the Von Neumann architecture and what data it stores during the fetch-execute cycle	
☹ 😐 ☺	I understand the difference between storing an **address and data**	
1.1.2 CPU Performance		
☹ 😐 ☺	I can explain how the **clock speed** affects the CPU performance	
☹ 😐 ☺	I can explain how the **cache size** affects the CPU performance	
☹ 😐 ☺	I can explain how the **number of cores** affects the CPU performance	
1.1.3 Embedded Systems		
☹ 😐 ☺	I can explain the purpose and **characteristics of embedded systems**	

☹ 😐 ☺	I can give a range of **examples of different embedded systems**	

1.2 Memory & Storage

Confidence	Clarification	✓
1.2.1 Primary storage (Memory)		
☹ 😐 ☺	I can explain the purpose and **characteristics of embedded systems**	
☹ 😐 ☺	I can give a range of **examples of different embedded systems**	
☹ 😐 ☺	I can explain why computers have **primary storage**	
☹ 😐 ☺	I can explain why primary storage usually consists of **ROM and RAM**	
☹ 😐 ☺	I can explain the **difference and key characteristics** of ROM and RAM	
☹ 😐 ☺	I can explain the **purpose of ROM** in a computer system	
☹ 😐 ☺	I can explain the **purpose of RAM** in a computer system	
☹ 😐 ☺	I can explain why **virtual memory** may be needed in a system	
☹ 😐 ☺	I can explain how **virtual memory** works and the transfer of data between RAM and the HDD	

Confidence	Clarification	✓
1.2.2 Secondary storage		
☹ 😐 ☺	I can explain the purpose and **characteristics of embedded systems**	
☹ 😐 ☺	I can explain why computers have **secondary storage**	
☹ 😐 ☺	I understand what **optical storage media** is and how it works	
☹ 😐 ☺	I can identify the advantages and disadvantages of **optical storage media**	
☹ 😐 ☺	I can identify different types of **optical storage** devices	

☹ 😐 ☺	I understand what **magnetic storage media** is and how it works	
☹ 😐 ☺	I can identify the advantages and disadvantages of **magnetic storage media**	
☹ 😐 ☺	I can identify different types of **magnetic storage** devices	
☹ 😐 ☺	I understand what **solid state storage media** is and how it works	
☹ 😐 ☺	I can identify the advantages and disadvantages of **solid state storage media**	
☹ 😐 ☺	I can identify different types of **solid state storage** devices	
☹ 😐 ☺	I can compare storage media and devices, identifying the most suitable for a given scenario	
☹ 😐 ☺	I can compare storage media based on their **capacity**	
☹ 😐 ☺	I can compare storage media based on their **speed**	
☹ 😐 ☺	I can compare storage media based on their **portability**	
☹ 😐 ☺	I can compare storage media based on their **durability**	
☹ 😐 ☺	I can compare storage media based on their **reliability**	
☹ 😐 ☺	I can compare storage media based on their **cost**	

10011
00111
11010

Confidence	**Clarification**	✓
1.2.3 Units		
☹ 😐 ☺	I can explain why data must be stored in **binary format**	
☹ 😐 ☺	I can explain how data needs to be converted into binary format to be processed by a computer	
☹ 😐 ☺	I can identify the different data units **bit, nibble (4 bits), byte (8 bits), kilobyte (1,000 bytes or 1KB), megabyte (1,000 KB), Gigabyte (1,000 MB), Terabyte (1,000 GB), Petabyte (1,000 TB)**	
☹ 😐 ☺	I can **calculate the capacity of devices** for different data units	
☹ 😐 ☺	I can **calculate the require storage capacity** for a given set of files	

☹ 😐 ☺	I can **calculate the file sizes** for sound files (file size = sample rate x duration(s) x bit depth)	
☹ 😐 ☺	I can **calculate the file sizes** for images (file size = colour depth x image height(px) x image width(px))	
☹ 😐 ☺	I can **calculate the file sizes** for text files (file size = bits per character x number of characters)	

1.2.4 Data Storage

Numbers

☹ 😐 ☺	I can **convert positive denary whole numbers (0 – 255) to binary numbers** (up to and including 8 bits) and vice versa	
☹ 😐 ☺	I can **add two binary integers together** (up to and including 8 bits) and **explain overflow errors** which may occur	
☹ 😐 ☺	I can **convert positive denary whole numbers** into **2-digit hexadecimal numbers** (00 – FF) and vice versa	
☹ 😐 ☺	I can **convert binary integers** to their **hexadecimal** equivalents and vice versa	
☹ 😐 ☺	I understand the terms '**most significant bit'**, and '**least significant bit'**.	
☹ 😐 ☺	I can carry out a **binary shift** and understand the effect (both left and right) on a number	

Characters

☹ 😐 ☺	I understand how binary codes are used to **represent characters**	
☹ 😐 ☺	I can explain what is meant by the term **'character set'**	
☹ 😐 ☺	I can explain the relationship between the **number of bits per character**, and the **number of characters** which can be represented, e.g. **ASCII and Unicode**	
☹ 😐 ☺	I can explain the **differences between and impact** of each character set	
☹ 😐 ☺	I understand how **character sets are logically ordered**, e.g. the code for 'B' will be one more than 'A'	
☹ 😐 ☺	I can perform a **binary shift** and understand the effect (both left and right) on a number	
☹ 😐 ☺	I know that the **binary representation of ASCII** in the exam will use 8 bits	

Confidence	**Clarification**	✓

1.2.4 Data Storage

Images

☹ 😐 ☺	I understand how an image is **represented as a series of pixels**, represented in binary	
☹ 😐 ☺	I can explain how each pixel has a **specific colour**, represented by a specific code	
☹ 😐 ☺	I can explain the **effect on image size and quality** when changing colour depth and resolution	
☹ 😐 ☺	I understand what additional information is **stored in the metadata** (e.g. height, width)	
Sound		
☹ 😐 ☺	I understand how analogue sound can be **sampled and stored** in digital form (binary).	
☹ 😐 ☺	I understand that the **sample rate** is measured in Hertz (Hz)	
☹ 😐 ☺	I understand that the **bit depth** refers to the number of bits available to store each sample (e.g. 16-bit)	
☹ 😐 ☺	I understand that the **duration** refers to the number of seconds of audio the sound file contains.	
☹ 😐 ☺	I can explain the effect of sample rate, duration and bit depth on the **playback quality** and **size of the sound file**.	
1.2.5 Compression		
☹ 😐 ☺	I understand the **need for compression**.	
☹ 😐 ☺	I can identify the advantages and disadvantages **lossy and lossless compression**.	
☹ 😐 ☺	I can explain the **effects on the file** for each type of compression.	
☹ 😐 ☺	I can identify **common scenarios** where compression may be needed.	

1.3 Computer networks, connections and protocols

Confidence	**Clarification**	✓
1.3.1 Networks and topologies		
☹ 😐 ☺	I can explain the difference between a **LAN (Local Area Network)** and **WAN (Wide Area Network)**	
☹ 😐 ☺	I understand the different **factors that can affect the performance** of a network, e.g. the numbers of devices connected and the bandwidth	
☹ 😐 ☺	I can explain the different roles of computers in a **client-server and a peer-to-peer** network	

☹ 😐 ☺	I can identify the hardware needed to **connect stand-alone computers** into a Local Area Network including **wireless access points, routers, switches, NIC (Network Interface Controller/Card) and Transmission media**.	
☹ 😐 ☺	I can explain the **tasks performed** by each piece of network hardware.	
☹ 😐 ☺	I understand the concept of the **Internet as a network of computer networks**.	
☹ 😐 ☺	I understand a **Domain Name Service (DNS)** is made up of multiple Domain Name Servers	
☹ 😐 ☺	I can explain the role of the **DNS (Domain Name Server)** in the conversion of a URL to an IP address.	
☹ 😐 ☺	I understand the **concept of servers providing services** (e.g. Web Server -> Web pages, File Server -> files storage/retrieval)	
☹ 😐 ☺	I understand the **concept of Hosting and clients requesting/using services** from a server	
☹ 😐 ☺	I understand what is meant by **'The Cloud'** and remote service provision (e.g. storage, software, processing)	
☹ 😐 ☺	I can explain the advantages and disadvantages of **'The Cloud'**.	
☹ 😐 ☺	I can explain the advantages and disadvantages of the **Star and Mesh** topologies	
☹ 😐 ☺	I can apply my understanding of **networks to a given scenario**	

1.3.2 Wired and wireless networks, protocols and layers

☹ 😐 ☺	I can compare the benefits and drawbacks of **wired (Ethernet) and wireless (Wi-Fi, Bluetooth) connections**	
☹ 😐 ☺	I am able to **recommend one or more connections** for a given scenario	
☹ 😐 ☺	I understand the **principle of encryption** to secure data across network connections	
☹ 😐 ☺	I understand **IP addressing** and the format of an IP address (IPv4 and IPv6) for communication over the Internet	
☹ 😐 ☺	I understand that **MAC addresses** are assigned to a devices for use within a network	
☹ 😐 ☺	I understand the principle of a **standard** to provide rules for areas of computing	
☹ 😐 ☺	I can explain how standards allow hardware/software to **interact across different manufacturers/producers**	
☹ 😐 ☺	I can explain the **principle of a (communication) protocol** as a set of rules for transferring data	

☹ 😐 ☺	I understand that different types of protocols are used for **different purposes**	
☹ 😐 ☺	I can explain the basic principle of the **TCP/IP (Transmission Control Protocol/Internet Protocol)** including its purpose and key features	
☹ 😐 ☺	I can explain the basic principle of the **HTTP (Hypertext Transfer Protocol)** including its purpose and key features	
☹ 😐 ☺	I can explain the basic principle of the **HTTPS (Hypertext Transfer Protocol Secure)** including its purpose and key features	
☹ 😐 ☺	I can explain the basic principle of the **FTP (File Transfer Protocol)** including its purpose and key features	
☹ 😐 ☺	I can explain the basic principle of the **POP (Post Office Protocol)** including its purpose and key features	
☹ 😐 ☺	I can explain the basic principle of the **IMAP (Internet Message Access Protocol)** including its purpose and key features	
☹ 😐 ☺	I can explain the basic principle of the **SMTP (Simple Mail Transfer Protocol)** including its purpose and key features	
☹ 😐 ☺	I understand the **concept of layers** and can explain the **benefits of using layers** e.g. the **4-layer TCP/IP model**	

1.4 Network Security

<u>Confidence</u>	<u>Clarification</u>	✓
1.4.1 Threats to computer systems and networks		
☹ 😐 ☺	I can explain how **Malware** is used, its purpose and the threats posed to devices/systems	
☹ 😐 ☺	I can explain how **Social engineering (e.g. phishing, people as a 'weak point')** is used, its purpose and the threats posed to devices/systems	
☹ 😐 ☺	I can explain how a **Brute-force attack** is used, its purpose and the threats posed to devices/systems	
☹ 😐 ☺	I can explain how a **Denial of service attack** is used, its purpose and the threats posed to devices/systems	
☹ 😐 ☺	I can explain how a **Data interception and theft** is used, its purpose and the threats posed to devices/systems	
☹ 😐 ☺	I can explain the concept of how **SQL injection** is used, its purpose and the threats posed to devices/systems	
1.4.2 Identifying and preventing vulnerabilities		
☹ 😐 ☺	I understand how to limit/prevent threats posed and remove vulnerabilities through use of **Penetration testing**	

☹ 😐 ☺	I understand how to limit/prevent threats posed and remove vulnerabilities through use of **Anti-malware software**	
☹ 😐 ☺	I understand how to limit/prevent threats posed and remove vulnerabilities through use of **Firewalls**	
☹ 😐 ☺	I understand how to limit/prevent threats posed and remove vulnerabilities through use of **User access levels**	
☹ 😐 ☺	I understand how to limit/prevent threats posed and remove vulnerabilities through use of **Passwords**	
☹ 😐 ☺	I understand how to limit/prevent threats posed and remove vulnerabilities through use of **Encryption**	
☹ 😐 ☺	I understand how to limit/prevent threats posed and remove vulnerabilities through use of **Physical security**	

1.5 Systems Software

Confidence	Clarification	✓
1.5.1 Operating systems		
☹ 😐 ☺	I can explain the purpose of the **operating system**	
☹ 😐 ☺	I can identify the different types and features of a **user interface**	
☹ 😐 ☺	I can explain how the operating system performs **memory management**, e.g. the transfer of data between memory, and how this allows for **multitasking**	
☹ 😐 ☺	I understand that **data is transferred** between devices and the processor, and that this process needs to be managed	
☹ 😐 ☺	I can explain how **peripherals** are managed and the need for **device drivers**	
☹ 😐 ☺	I can explain **user management functions**, e.g. allocation of an account, access rights, security, etc.	
☹ 😐 ☺	I can explain **file management**, and the key features, e.g. naming, allocating to folders, moving files, saving, etc.	
1.5.2 Utility software		
☹ 😐 ☺	I understand that computers often come with **utility software**, and how this performs housekeeping tasks	
☹ 😐 ☺	I can explain the purpose of **encryption** software and why it is required	
☹ 😐 ☺	I can explain the purpose of **defragmentation** software and why it is required	

☹ 😐 ☺	I can explain the purpose of **data compression** software and why it is required	

1.6 Ethical, legal, cultural and environmental concerns

Confidence	Clarification	✓
1.6.1 Ethical, legal, cultural and environmental impact		
☹ 😐 ☺	I understand that technology introduces **ethical, legal, cultural, environmental and privacy issues**.	
☹ 😐 ☺	I can discuss, using a variety of digital technology examples, **ethical issues** and the impact of technology on wider society	
☹ 😐 ☺	I can discuss, using a variety of digital technology examples, **privacy issues** and the impact of technology on wider society	
☹ 😐 ☺	I can discuss, using a variety of digital technology examples, **cultural issues** and the impact of technology on wider society	
☹ 😐 ☺	I can discuss, using a variety of digital technology examples, **environmental issues** and the impact of technology on wider society	
☹ 😐 ☺	I can discuss, using a variety of digital technology examples, **legal issues** and the impact of technology on wider society	
☹ 😐 ☺	I can explain the purpose of **The Data Protection Act 2018** and the specific actions it allows or prohibits	
☹ 😐 ☺	I can explain the purpose of the **Computer Misuse Act 1998** and the specific actions it allows or prohibits	
☹ 😐 ☺	I can explain the purpose of the **Copyright Designs and Patents Act 1988** and the specific actions it allows or prohibits	
☹ 😐 ☺	I can explain the need to license software and the purpose of a **software license**	
☹ 😐 ☺	I can explain the features of **open source software** (providing access to the source code and the ability to change the software)	
☹ 😐 ☺	I can explain the features of **proprietary software** (no access to the source code, purchased commonly as off-the-shelf)	
☹ 😐 ☺	I can **recommend a type of license** for a given scenario including benefits and drawbacks	

2.1 Algorithms

Confidence	Clarification	✓
2.1.1 Computational Thinking		
☹ 😐 ☺	I understand how the principle of **Abstraction** and how it can be used to define and refine problems	
☹ 😐 ☺	I understand how the principle of **Decomposition** and how it can be used to define and refine problems	
☹ 😐 ☺	I understand how the principle of **Algorithmic Thinking** and how it can be used to define and refine problems	
2.1.2 Designing, creating and refining algorithms		
☹ 😐 ☺	I can identify the **inputs**, **processes**, and **outputs** for a problem	
☹ 😐 ☺	I produce simple **structure diagrams** to show the structure of a problem, the subsections and their links to other subsections	
☹ 😐 ☺	I can complete, write or refine an algorithm using **pseudocode**	
☹ 😐 ☺	I can complete, write or refine an algorithm using **flowcharts**	
☹ 😐 ☺	I can complete, write or refine an algorithm using **OCR reference language** or **Python**	
☹ 😐 ☺	I can identify **syntax/logic errors** in code and suggest fixes	
☹ 😐 ☺	I can create and use **trace tables** to follow an algorithm	
2.1.3 Searching and sorting algorithms		
☹ 😐 ☺	I understand the main steps of a **binary search** algorithm	
☹ 😐 ☺	I understand the pre-requisites of a **binary search** algorithm	
☹ 😐 ☺	I can apply a **binary search** algorithm to a data set	
☹ 😐 ☺	I can identify a **binary search** algorithm if given the code or pseudocode for it	

☹ 😐 ☺	I understand the main steps of a **linear search** algorithm	
☹ 😐 ☺	I can apply a **linear search** algorithm to a data set	
☹ 😐 ☺	I can identify a **binary search** algorithm if given the code or pseudocode for it	
☹ 😐 ☺	I understand the main steps of a **bubble sort** algorithm	
☹ 😐 ☺	I can apply a **bubble sort** algorithm to a data set	
☹ 😐 ☺	I can identify a **bubble sort** algorithm if given the code or pseudocode for it	
☹ 😐 ☺	I understand the main steps of a **merge sort** algorithm	
☹ 😐 ☺	I can apply a **merge sort** algorithm to a data set	
☹ 😐 ☺	I can identify a **merge sort** algorithm if given the code or pseudocode for it	
☹ 😐 ☺	I understand the main steps of a **insertion sort** algorithm	
☹ 😐 ☺	I can apply a **insertion sort** algorithm to a data set	
☹ 😐 ☺	I can identify a **insertion sort** algorithm if given the code or pseudocode for it	

2.2 Programming fundamentals

Confidence	Clarification	✓
2.2.1 Programming fundamentals		
☹ 😐 ☺	I understand and can use **variables** and **constants** in the Python high-level programming language	
☹ 😐 ☺	I understand and can use **inputs** and **outputs** in the Python high-level programming language	
☹ 😐 ☺	I understand and can use the **comparison operators** (**==** Equal to, **!=** Not equal to, **<** Less than, **<=** Less than or equal to, **>** Greater than, **>=** Greater than or equal to)	
☹ 😐 ☺	I understand and can use the **arithmetic operators** (**+** Addition, **-** Subtraction, ***** Multiplication, **/** Division, **MOD** Modulus, **DIV** Quotient, **^** Exponentiation (to the power))	
☹ 😐 ☺	I understand the use of the **three programming constructs (sequence, selection, and iteration)** and how they're used to control the flow of a program	
☹ 😐 ☺	I am able to identify **sequence** in code or pseudocode	
☹ 😐 ☺	I am able to identify **selection** in code or pseudocode	

☹ 😐 ☺	I am able to identify **iteration (count and condition controlled loops)** in code or pseudocode	
☹ 😐 ☺	I understand and can use the **Boolean operators AND, OR** and **NOT**	
2.2.2 Data types		
☹ 😐 ☺	I understand and can use **Integer** numbers in the Python high-level programming language	
☹ 😐 ☺	I understand and can use **Real** numbers in the Python high-level programming language	
☹ 😐 ☺	I understand and can use **Boolean** values in the Python high-level programming language	
☹ 😐 ☺	I understand and can use **Characters and Strings** in the Python high-level programming language	
☹ 😐 ☺	I understand and can use **Casting** in the Python high-level programming language	
☹ 😐 ☺	I am able to choose **suitable data types** for data in a given scenario	
☹ 😐 ☺	I understand that data types may be **temporarily changed through casting**, and where this may be useful	
2.2.3 Additional programming techniques		
☹ 😐 ☺	I understand and can use **basic string manipulation (Concatenation and Slicing)** in the Python high-level programming language	
☹ 😐 ☺	I understand and can use **basic file handling operations (Open, Read, Write, and Close)** in the Python high-level programming language	
☹ 😐 ☺	I understand the use of **records** to store data	
☹ 😐 ☺	I understand the use of **arrays** as fixed length or static structures when solving problems, including both **one-dimensional (1D)** and **two-dimensional arrays (2D)**	
☹ 😐 ☺	I understand the use of **2D arrays** to emulate database tables of a collection of fields, and records.	
☹ 😐 ☺	I understand the use of **SQL** to search for data in a database	
☹ 😐 ☺	I understand and can write SQL statements using **SELECT, FROM** and **WHERE**	
☹ 😐 ☺	I understand how to use sub programs (**functions and procedures**) to produce structured code and where to use them effectively	
☹ 😐 ☺	I can use **local variables/constants** within functions and procedures	
☹ 😐 ☺	I can use **global variables/constants** within functions and procedures	
☹ 😐 ☺	I can use **arrays (passing and returning)** within functions and procedures	

☹ 😐 ☺	I can create and use **random numbers** in a program	

2.3 Producing robust programs

Confidence	Clarification	✓
2.3.1 Defensive design		
☹ 😐 ☺	I understand the issues a programmer should consider to ensure that a program caters for **all likely input values** and **anticipate misuse**	
☹ 😐 ☺	I understand how to deal with **invalid data** in a program	
☹ 😐 ☺	I understand the **authentication** is required to confirm the identity of a user	
☹ 😐 ☺	I am able to design and create python code to provide **simple authentication** of a user (e.g. username and password)	
☹ 😐 ☺	I understand why **commenting code** is useful and can apply this appropriately	
☹ 😐 ☺	I understand the benefits of using **sub programs** to improve maintainability	
☹ 😐 ☺	I understand the benefits of using **naming conventions** for variables and constants to improve maintainability	
☹ 😐 ☺	I understand the benefits of **indentation** in program code to improve maintainability	
2.3.2 Testing		
☹ 😐 ☺	I understand the **purpose of testing**	
☹ 😐 ☺	I can explain the difference between **Iterative** (testing modules of a program during development) and **Final/terminal testing** (testing the program at the end of production)	
☹ 😐 ☺	I can identify **Syntax errors** as errors which break the grammatical rules of the programming language and stop it from being run/translated	
☹ 😐 ☺	I can identify **Logic errors** as errors which produce unexpected output	
☹ 😐 ☺	I can identify **Normal test data** as data which should be accepted by a program without causing errors	
☹ 😐 ☺	I can identify **Boundary test data** as data which should be accepted by a program without causing errors	
☹ 😐 ☺	I can identify **Invalid test data** as data of the correct data type which should be rejected by the computer system	
☹ 😐 ☺	I can identify **Erroneous test data** as data of the incorrect data type which should be rejected by the computer system	

☹ 😐 ☺	I am able to identify **suitable test data** for a given scenario	
☹ 😐 ☺	I am able to create and complete a **test plan**	
☹ 😐 ☺	I am able to use **test results** to refine algorithms	

2.4 Boolean logic

Confidence	Clarification	✓
2.4.1 Boolean logic		
☹ 😐 ☺	I can identify the **logic gate symbols** for AND, OR and NOT	
☹ 😐 ☺	I understand how each **logic gate works** and can complete truth tables for them	
☹ 😐 ☺	I can complete simple **logic gate diagrams** using the operators AND, OR and NOT	
☹ 😐 ☺	I can combine **Boolean operators** using AND, OR and NOT	
☹ 😐 ☺	I can create, complete or edit **logic diagrams** for given scenarios	
☹ 😐 ☺	I can create, complete or edit **truth tables** for given scenarios	
☹ 😐 ☺	I can apply **logical operators in truth tables** to solve problems	
☹ 😐 ☺	I am able to use **test results** to refine algorithms	

2.5 Programming languages and IDE’s

Confidence	Clarification	✓
2.5.1 Languages		
☹ 😐 ☺	I can explain the differences between **High and Low level programming languages**	
☹ 😐 ☺	I can explain the purpose of **Translators**	
☹ 😐 ☺	I can explain the differences, benefits and drawbacks of using a **Compiler** or an **Interpreter**	
2.5.2 The Integrated Development Environment (IDE)		

☹ 😐 ☺	I can identify the **tools** that an IDE provides	
☹ 😐 ☺	I can explain how **Editors** can be used to help a programmer develop a program	
☹ 😐 ☺	I can explain how **Error diagnostics** can be used to help a programmer develop a program	
☹ 😐 ☺	I can explain how a **Run-time environment** can be used to help a programmer develop a program	
☹ 😐 ☺	I can explain how **Translators** can be used to help a programmer develop a program	
☹ 😐 ☺	I have **practical experience** of using a range of these tools within an IDE	