Name:

Unit 1 Revision Checklist

1.1 Systems Architecture	☺	☹
The purpose of the CPU		
CPU components and their function:		
• The Control Unit (CU)		
• The Arithmetic Logic Unit (ALU)		
• Cache		
How a CPU carries out instructions using the; Fetch-Decode-Execute Cycle stored in memory		
Von Neumann architecture:		
• Memory Address Register (MAR)		
• Memory Data Register (MDR)		
• Program Counter (PC)		
• Accumulator		
How common characteristics of CPU's affect their performance:		
• clock speed		
• cache size		
• number of cores		
Embedded systems:		
• the purpose of embedded systems		
• examples of embedded systems		

1.2 Memory	☺	☹
The difference between RAM and ROM		
The purpose of ROM in a computer system		
The purpose of RAM in a computer system		
The need for virtual memory		
Flash memory		

1.3 Storage	☺	☹
The three tiers of storage:		
• Primary storage		
• Secondary storage		
• Tertiary storage		
The need for secondary storage		
Data capacity and calculation of data capacity requirements		
Common types of storage		
• Optical		

Name:

• Magnetic • Solid State		
Suitable storage devices and storage media for a given application, the advantages and disadvantages of these, using characteristics:		
• Capacity		
• Speed		
• Portability		
• Durability		
• Reliability		
• Cost		

1.4 Wired and Wireless Networks	☺	☹
Types of network:		
• Local Area Network (LAN)		
• Wide Area Network (WAN)		
Factors that affect the performance of networks		
The different roles of computers in a client-server and peer-to-peer network		
The hardware needed to connect stand-alone computers into a Local Area Network:		
• Wireless access point (WAP)		
• Routers/ switches		
• Network Interface Controller/Card (NIC)		
• Transmission media		
The internet as a worldwide collection of computer networks:		
• Domain Name Server (DNS)		
• Hosting		
• The Cloud		
The concept of virtual networks		

1.5 Network topologies, protocols and layers	☺	☹
Star and mesh network topologies		
WiFi:		
• Frequency and channels		
• Encryption		
Ethernet		
The uses of IP addressing, MAC addressing and protocols including:		
• Transmission Control Protocol/ Internet Protocol TCP/IP		
• Hyper Text Transfer Protocol (HTTP)		
• Hyper Text Transfer Protocol Secure (HTTPS)		
• FTP (File Transfer Protocol)		
• Post Office Protocol (POP)		

Name:

• Internet Message Access Protocol (IMAP) • Simple Mail Transfer Protocol (SMTP)		
The concept of layers		
Packet switching		

1.6 System Security	☺	☹
Forms of attack		
Threats posed to networks:		
• Malware		
• Phishing		
• People as the ‘weak point’ in secure systems (social engineering)		
• Brute force attacks		
• Denial of service attacks		
• Data interception and theft		
• The concept of SQL injection		
• Poor network policy		
Identifying and preventing vulnerabilities		
• Penetration testing		
• Network forensics		
• Network policies		
• Anti-malware software		
• Firewalls		
• User access levels		
• Passwords		
• Encryption		

1.7 Systems Software	☺	☹
The purpose and functionality of systems software		
Operating systems:		
• User interface		
• Memory management/ multitasking		
• Peripheral management and drivers		
• User management		
• File management		
Utility system software:		
• Encryption software		
• Defragmentation		
• Data compression		
• The role and methods of backup:		
○ Full		
○ Incremental		

Name:

1.8 Ethical, Legal, Cultural and environmental concerns	☺	☹
How to investigate and discuss Computer Science technologies while considering:		
• Ethical issues		
• Legal issues		
• Cultural issues		
• Environmental issues		
• Privacy issues		
How key stakeholders are affected by technologies		
Environmental impact of Computer Science		
Cultural implications of Computer Science		
Open Source vs Proprietary Software		
Legislation relevant to Computer Science:		
• The Data Protection Act 1998		
• Computer Misuse Act 1990		
• Copyright, Designs and Patents Act 1988		
• Creative Commons Licensing		
• Freedom of Information Act 2000		

Name:

Unit 2 Revision Checklist

2.1 Algorithms	☺	☹
Computational thinking		
• abstraction		
• decomposition		
• algorithmic thinking		
Standard searching algorithms:		
• binary search		
• linear search		
Standard sorting algorithms:		
• bubble sort		
• merge sort		
• insertion sort		
How to produce algorithms using:		
• pseudocode		
• using flow diagrams		
Interpret, correct or complete algorithms		

2.2 Programming Techniques	☺	☹
The use of:		
• variables		
• constants		
• operators		
• inputs		
• outputs		
• assignments		
The use of three basic programming constructs used to control the flow of a program:		
• sequence		
• selection		
• iteration (count and condition controlled loops)		
The use of basic string manipulation		
The use of basic file handling operation:		
• open		
• read		
• write		
• close		
The use of records to store data		
The use of SQL to search for data		
The use of arrays (or equivalent) when solving problems, including:		
• one dimensional arrays		
• two dimensional arrays		

Name:

How to use sub programs (functions and procedures) to produce structured code		
The use of data types:		
• integer		
• real		
• Boolean		
• character and string		
• casting		
The common arithmetic operators		
The common Boolean operators		

2.3 Producing robust programs	☺	☹
Defensive design considerations:		
• input sanitisation/ validation		
• planning for contingencies		
• anticipating misuse		
• authentication		
Maintainability:		
• comments		
• indentation		
The purpose of testing		
Types of testing:		
• iterative		
• final/ terminal		
How to identify syntax and logic errors		
Selecting and using suitable test data		

2.4 Computational Logic	☺	☹
Why data is represented in computer systems in binary form		
Simple logic diagrams using the operations AND, OR and NOT		
Truth tables		
Combining Boolean operators using AND, OR and NOT to two levels		
Applying logical operators in appropriate truth tables to solve problems		
Applying computing-related mathematics:		
• +		
• -		
• /		
• *		
• Exponentiation (^)		
• MOD		
• DIV		

Name:

2.5 Translators and facilities of languages	☺	☹
Characteristics and purpose of different levels of programming language, including low level languages		
The purpose of translators		
The characteristics of an assembler, a compiler and an interpreter		
Common tools and facilities available in an integrated development environment (IDE):		
• editors		
• error diagnostics		
• run-time environment		
• translators		

2.6 Data representation	☺	☹
Units:		
• bit, nibble, byte, kilobyte, megabyte, gigabyte, terabyte, petabyte		
• how data needs to be converted into a binary format to be processed by a computer.		
Numbers:		
• how to convert positive denary whole numbers (0-255) into 8 bit binary numbers and vice versa		
• how to add two 8 bit binary integers and explain overflow errors which may occur		
• binary shifts		
• how to convert positive denary whole numbers (0-255) into 2 digit hexadecimal numbers and vice versa		
• how to convert from binary to hexadecimal equivalents and vice versa		
• check digits		
Characters:		
• the use of binary code to represent characters		
• the term ‘character set’		
• the relationship between the number of bits per character in a character set and the number of characters which can be represented (for example ASCII, extended ASCII and Unicode)		
Images:		
• how and image is represented as a series of pixels represented in binary		
• metadata included in the file		
• the effect of colour depth and resolution on the size of an image file.		

Name:

Sound:		
• how sound can be sampled and stored in digital form		
• how sampling intervals and other factors affect the size of a sound file and the quality of its playback:		
○ sample size		
○ bit rate		
○ sampling frequency		
Compression:		
• need for compression		
• types of compression		
○ lossy		
○ lossless		