  
 **Student Learning Reflection & Personalised Learning Checklist**

|  |  |
| --- | --- |
| **Subject/Course:** | **Computer Science GCSE** |
| **Student Name:** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Self Assessment | | |
| Topic | Key knowledge/skills | Red | Amber | Green |
| **Paper 1** | **Computer Systems** |  |  |  |
| **1.1 Systems architecture** | The purpose of the CPU   * The fetch-execute cycle |  |  |  |
|  | Common CPU components and their function:   * ALU (Arithmetic Logic Unit) * CU (Control Unit) * Cache * Registers |  |  |  |
|  | Von Neumann architecture:   * MAR (Memory Address Register) * MDR (Memory Data Register) * Program Counter * Accumulator |  |  |  |
|  | How common characteristics of CPUs affect their performance:   * Clock speed * Cache size * Number of cores |  |  |  |
|  | * The purpose and characteristics of embedded systems * Examples of embedded systems |  |  |  |
| **1.2 Memory and Storage** | * The need for primary storage * The difference between RAM and ROM * The purpose of ROM in a computer system * The purpose of RAM in a computer system * Virtual memory |  |  |  |
|  | * The need for secondary storage * Common types of storage:   + Optical   + Magnetic   + Solid state * Suitable storage devices and storage media for a given application * The advantages and disadvantages of different storage devices and storage media relating to these characteristics:   + Capacity   + Speed   + Portability   + Durability   + Reliability   + Cost |  |  |  |
|  | * The units of data storage:   + Bit   + Nibble (4 bits)   + Byte (8 bits)   + Kilobyte (1,000 bytes or 1 KB)   + Megabyte (1,000 KB)   + Gigabyte (1,000 MB)   + Terabyte (1,000 GB)   + Petabyte (1,000 TB) * How data needs to be converted into a binary format to be processed by a computer * Data capacity and calculation of data capacity requirements |  |  |  |
|  | **Numbers**   * How to convert positive denary whole numbers to binary numbers (up to and including 8 bits) and vice versa * How to add two binary integers together (up to and including 8 bits) and explain overflow errors which may occur * How to convert positive denary whole numbers into 2-digit hexadecimal numbers and vice versa * How to convert binary integers to their hexadecimal equivalents and vice versa * Binary shifts |  |  |  |
|  | **Characters**   * The use of binary codes 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, e.g.:   + ASCII   + Unicode   **Images**   * How an image is represented as a series of pixels, represented in binary * Metadata * The effect of colour depth and resolution on:   + The quality of the image   + The size of an image file   **Sound**   * How sound can be sampled and stored in digital form * The effect of sample rate, duration and bit depth on:   + The playback quality   + The size of a sound file |  |  |  |
|  | * The need for compression * Types of compression:   + Lossy   + Lossless |  |  |  |
| **1.3 Computer networks, connections and protocols** | * Types of network:   + LAN (Local Area Network)   + WAN (Wide Area Network) * Factors that affect the performance of networks * The different roles of computers in a client-server and a peer-to-peer network * The hardware needed to connect stand-alone computers into a Local Area Network:   + Wireless access points   + Routers   + Switches   + NIC (Network Interface Controller/Card)   + Transmission media * The Internet as a worldwide collection of computer networks:   + DNS (Domain Name Server)   + Hosting   + The Cloud   + Web servers and clients * Star and Mesh network topologies |  |  |  |
|  | * Modes of connection:   + Wired     - Ethernet   + Wireless     - Wi-Fi     - Bluetooth * Encryption * IP addressing and MAC addressing * Standards * Common protocols including:   + TCP/IP (Transmission Control Protocol/Internet Protocol)   + HTTP (Hyper Text Transfer Protocol)   + HTTPS (Hyper Text Transfer Protocol Secure)   + FTP (File Transfer Protocol)   + POP (Post Office Protocol)   + IMAP (Internet Message Access Protocol)   + SMTP (Simple Mail Transfer Protocol) * The concept of layers |  |  |  |
| **1.4 Network Security** | * Forms of attack:   + Malware   + Social engineering, e.g. phishing, people as the ‘weak point’   + Brute-force attacks   + Denial of service attacks   + Data interception and theft   + The concept of SQL injection |  |  |  |
|  | * Common prevention methods:   + Penetration testing   + Anti-malware software   + Firewalls   + User access levels   + Passwords   + Encryption   + Physical security |  |  |  |
| **1.5 Systems Software** | * The purpose and functionality of operating systems:   + User interface   + Memory management and multitasking   + Peripheral management and drivers   + User management   + File management |  |  |  |
|  | * The purpose and functionality of utility software * Utility system software:   + Encryption software   + Defragmentation   + Data compression |  |  |  |
| **1.6 Ethical, legal, cultural and environmental impacts of digital technology** | * Impacts of digital technology on wider society including:   + Ethical issues   + Legal issues   + Cultural issues   + Environmental issues   + Privacy issues * Legislation relevant to Computer Science:   + The Data Protection Act 2018   + Computer Misuse Act 1990   + Copyright Designs and Patents Act 1988   + Software licences (i.e. open source and proprietary) |  |  |  |
| **Paper 2** | **Computational Thinking, algorithms and programming** |  |  |  |
| **2.1 Algorithms** | computational thinking:   * abstraction * decomposition * algorithmic thinking |  |  |  |
|  | * Identify the inputs, processes, and outputs for a problem * Structure diagrams * Create, interpret, correct, complete, and refine algorithms using:   + Pseudocode   + Flowcharts   + Reference language/high-level programming language * Identify common errors * Trace tables |  |  |  |
|  | * Standard searching algorithms:   + Binary search   + Linear search * Standard sorting algorithms:   + Bubble sort   + Merge sort   + Insertion sort |  |  |  |
| **2.2 Programming fundamentals** | * The use of variables, constants, operators, inputs, outputs and assignments * The use of the three basic programming constructs used to control the flow of a program:   + Sequence   + Selection   + Iteration (count- and condition-controlled loops) * The common arithmetic operators * The common Boolean operators AND, OR and NOT |  |  |  |
|  | * The use of data types:   + Integer   + Real   + Boolean   + Character and string   + Casting |  |  |  |
|  | * The use of basic string manipulation * The use of basic file handling operations:   + 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 both one-dimensional (1D) and two-dimensional arrays (2D)   How to use sub programs (functions and procedures) to produce structured code   * Random number generation |  |  |  |
| **2.3 Defensive design** | * Defensive design considerations:   + Anticipating misuse   + Authentication * Input validation * Maintainability:   + Use of sub programs   + Naming conventions   + Indentation   + Commenting |  |  |  |
|  | * The purpose of testing * Types of testing:   + Iterative   + Final/terminal * Identify syntax and logic errors * Selecting and using suitable test data:   + Normal   + Boundary   + Invalid/Erroneous * Refining algorithms |  |  |  |
| **2.4 Boolean Logic** | * Simple logic diagrams using the operators AND, OR and NOT * Truth tables * Combining Boolean operators using AND, OR and NOT * Applying logical operators in truth tables to solve problems |  |  |  |
| **2.5 Programming languages and Integrated Development Environments** | * Characteristics and purpose of different levels of programming language:   + High-level languages   + Low-level languages * The purpose of translators * The characteristics of a compiler and an interpreter |  |  |  |
|  | * Common tools and facilities available in an Integrated Development Environment (IDE):   + Editors   + Error diagnostics   + Run-time environment   + Translators |  |  |  |