

**1.1 SYSTEMS ARCHITECTURE**

- The purpose of the CPU
- Von Neumann architecture:
  - MAR (Memory Address Register)
  - MDR (Memory Data Register)
  - Program Counter
  - Accumulator
- Common CPU components and their function:
  - ALU (Arithmetic Logic Unit)
  - CU (Control Unit)
  - Cache
- The function of the CPU as fetch and execute instructions stored in memory
- How common characteristics of CPU's affect their performance:
  - Clock speed
  - Cache size
  - Number of cores
- Embedded systems:
  - 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 need for secondary storage
- Data capacity and calculation of data capacity requirements
- Common types of storage:
  - Optical
  - Magnetic
  - Solid state
- Suitable storage devices and storage media for a given application, and the advantages and disadvantages of these, using characteristics:
  - Capacity
  - Speed
  - Portability
  - Durability
  - Reliability
  - Cost

**1.4 WIRED AND WIRELESS NETWORKS**

- Types of networks:
  - 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 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
- 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:
  - 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
- 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

### **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 v 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

### **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 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 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 and two dimensional arrays.

- 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
  - 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

### 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 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 (for example ASCII, extended ASCII and Unicode)
- Images
  - How an 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
- Sound
  - How sound can be sampled and stored in a 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