Computer System and Programming with Algorithms (32 Hours)

1.1 – Systems architecture (2.5 hours)

1.1.1 Architecture of the CPU

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

1.1.2 CPU performance

How common characteristics of CPUs affect their performance:

- Clock speed
- Cache size
- Number of cores

1.2 – Memory and storage (2.5 hours)

1.2.1 Primary storage (memory)

- 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
- Cache

1.2.2 Secondary storage

- 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

1.2.3 Units

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

1.2.4 Data storage

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.:

o ASCII

o Unicode

Images

How an image is represented as a series of pixels, represented in binary

Metadata

The effect of colour depth and resolution on:

o The quality of the image

o 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:

o The playback quality

o The size of a sound file

1.2.5 Compression

The need for compression

Types of compression:

o Lossy

o Lossless

1.3 – Computer networks, connections and protocols (4 hours)

1.3.1 Networks and topologies

Types of network:

o LAN (Local Area Network)

o WAN (Wide Area Network)

Factors that affect the performance of networks

The different roles of computers in a client-server and a peer-topeer network

The hardware needed to connect stand-alone computers into a

Local Area Network:

o Wireless access points

o Routers

o Switches

o NIC (Network Interface Controller/Card)

o Transmission media

The Internet as a worldwide collection of computer networks:

o DNS (Domain Name Server)

o Hosting

o The Cloud

o Web servers and clients

Star and Mesh network topologies

1.3.2 Wired and wireless networks, protocols and layers

Modes of connection:

o Wired

• Ethernet

o Wireless

• Wi-Fi

• Bluetooth

Encryption

IP addressing and MAC addressing

Standards

Common protocols including:

o TCP/IP (Transmission Control Protocol/Internet Protocol)

o HTTP (Hyper Text Transfer Protocol)

o HTTPS (Hyper Text Transfer Protocol Secure)

o FTP (File Transfer Protocol)

o POP (Post Office Protocol)

o IMAP (Internet Message Access Protocol)

o SMTP (Simple Mail Transfer Protocol)

The concept of layers

1.4 – Network security (2 hours)

1.4.1 Threats to computer systems and networks

Forms of attack:

o Malware

o Social engineering, e.g. phishing, people as the 'weak point'

o Brute-force attacks

o Denial of service attacks

o Data interception and theft

o The concept of SQL injection

1.4.2 Identifying and preventing vulnerabilities

Common prevention methods:

o Penetration testing

o Anti-malware software

o Firewalls

o User access levels

o Passwords

o Encryption

o Physical security

1.5 – Systems software (2 hours)

1.5.1 Operating systems

The purpose and functionality of operating systems:

o User interface

o Memory management and multitasking

o Peripheral management and drivers

o User management

o File management

1.5.2 Utility software

The purpose and functionality of utility software

Utility system software:

o Encryption software

o Defragmentation

o Data compression

2.1 – Algorithms (4 hours)

2.1.1 Computational thinking

Principles of computational thinking:

o Abstraction

o Decomposition

o Algorithmic thinking

2.1.2 Designing, creating and refining algorithms

Identify the inputs, processes, and outputs for a problem

Structure diagrams

Create, interpret, correct, complete, and refine algorithms using:

o Pseudocode

o Flowcharts

o Reference language/high-level programming language

Identify common errors

Trace tables

2.1.3 Searching and sorting algorithms

Standard searching algorithms:

o Binary search

o Linear search

Standard sorting algorithms:

o Bubble sort

o Merge sort

o Insertion sort

2.2 – Programming fundamentals (10 hours)

2.2.1 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:

o Sequence

o Selection

o Iteration (count- and condition-controlled loops)

The common arithmetic operators

The common Boolean operators AND, OR and NOT

2.2.2 Data types

The use of data types:

o Integer

o Real

o Boolean

o Character and string

o Casting

2.2.3 Additional programming techniques

The use of basic string manipulation

The use of basic file handling operations:

o Open

o Read

o Write

o 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 – Producing robust programs (3 hours)

2.3.1 Defensive design

Input validation

Maintainability:

o Use of sub programs

o Naming conventions

o Indentation

o Commenting

2.3.2 Testing

The purpose of testing

Types of testing:

o Iterative

o Final/terminal

Identify syntax and logic errors

Selecting and using suitable test data:

o Normal

o Boundary

o Invalid/Erroneous

Refining algorithms

2.5 – Programming languages and Integrated Development Environments (2 hours)

2.5.1 Languages

Characteristics and purpose of different levels of programming

language:

o High-level languages

o Low-level languages

The purpose of translators

The characteristics of a compiler and an interpreter

2.5.2 The Integrated Development Environment (IDE)

Common tools and facilities available in an Integrated

Development Environment (IDE):

- Editors
- Error diagnostics
- Run-time environment
- Translators