Revision List for Year 11 Computer Science

This a comprehensive list of everything you have studied on the course so far.

Paper 1 Computer Systems

1.1.1 Architecture of the CPU

- ❑ The purpose of the CPU
  - The fetch-execute cycle
- ❑ 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
- ❑ Registers

1.1.2 CPU performance

- ❑ How common characteristics of CPUs affect their performance:
  - Clock speed
  - Cache size
  - Number of cores

1.1.3 Embedded systems:

- ❑ The purpose and characteristics of embedded systems
- ❑ Examples of embedded systems.

1.2 Memory and Storage

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
- ❑ The need for virtual memory

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 1KB)
  - 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 calculations 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.

- ASCII
- Unicode

1.2.5 Compression

❑ The need for compression

❑ Types of compression:

- Lossy
- Lossless

1.6. Ethical, legal, cultural and environmental concerns

❑ 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 1998
- Computer Misuse Act 1990
- Copyright Designs and Patents Act 1988
- Software licences (ie open source and proprietary)

Paper 2 – Computational Thinking, Algorithms and Programming

2.1.1 Computational Thinking

- ❑ Principles of computational thinking:
  - abstraction
  - decomposition
  - 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
  - pseudocode
  - flowcharts
  - reference language/high-level programming language
- ❑ Identify common errors
- ❑ Trace tables

2.1.3 Searching and sorting algorithms

- ❑ Standard searching algorithms:
  - binary search
  - linear search
- ❑ Standard sorting algorithms:
  - bubble sort
  - merge sort
  - insertion sort

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:
  - sequence
  - selection
  - iteration (count and condition-controlled loops)
- ❑ The common arithmetic operators

- ❏ The common Boolean operators AND, OR, NOT.

2.2.2 Data types

- ❏ The use of data types:
  - integer
  - real
  - Boolean
  - character and string
  - casting

2.3.2 Testing

- ❏ 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

2.4.1 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

2.5.1 Languages

- ❏ 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

2.5.2 The Integrated Development Environment (IDE)

- ❑ Common tools and facilities available in an IDE:
  - editors
  - error diagnostics
  - run-time environment
  - translators