End of Year Exams: Revision Guidance

Circulation Year 10 Students
Title Yr10 End of Year Exam Revision
Purpose To provide revision information for End of Year Exams

Year 10

Exam Dates:

22nd June - Paper 1 (90 mins)
23rd June - Paper 2 (90 mins)

Please find attached the revision list for you end of year exam, please use the following revision resources that have been set on classcharts:

- Resources in the Files area of the 10B Microsoft Team

- PLCs from previous tests

- Revision Guides

- Smart Revise (Make sure you filter by topic - not every topic is in your exam)

- GCSE Specification (attached) so you are aware of precisely what you do and don't need to know.

- Key Terms (attached)

REVISION LIST PAPER 1

- **Units of Measurement** (Bit, Byte, Kilobyte, Megabyte, Gigabyte, Terrabyte)
- **Data Representation** – Converting Binary to Denary, Converting Binary to Denary, Binary Addition**,** Converting Hexadecimal to Denary, Converting Hexadecimal to Binary, Converting Binary to Hexadecimal, Converting Denary to Hex, Converting Hex to Denary.
- **Data Representation** – Understanding how Images, Sound and Characters are stored as Binary:
    - Images (Resolution, Colour Depth)
    - Sound (Sample Rate, Bit Depth)
    - Logic Gates (AND, OR, NOT gates)
    - Drawing logic circuits
    - Completing truth tables
- **Algorithms**
    - Understanding and recognizing the terms: Sequence, Selection (IF/ELSE statements) and Iteration (for loops and while loops)
    - Creating algorithms/programs to solve problems.
    - Tracing algorithms/programs to solve problems.
- **Networks**
    - Types of Network (LAN and WAN)
    - Factors affecting network performance
    - Types of network (Client Server and Peer to Peer Networks)
    - The Internet (Remote Servers, Clients and the Cloud)
    - DNS
    - Star and Mesh Topologies
    - Encryption

- Mac and IP addresses
- Network Protocols and Standards
- The concept of layers

- **Hardware**
  - Understanding and recognizing the difference between hardware and software.
  - Understanding the purpose of the CPU and different hardware
  - Memory – Understanding RAM, ROM, Virtual Memory and Cache memory
  - Storage – Understanding secondary storage and the different characteristics when choosing secondary storage devices.

REVISION LIST PAPER 2

- **Designing, creating and refining algorithms**
  - Identify the inputs, processes, and outputs for a problem
  - Using either
    - a flow chart with the correct symbols
    - High Level Language (python) or exam reference language or pseudocode
  - Identify syntax/logic errors in code and suggest fixes
  - Use of variables, constants, operators, inputs, outputs and assignments

- **Testing**
  - Understanding types of testing (iterative and final)
  - Understanding types of test data (normal, boundary and invalid/erroneous data)
  - Be able to suggest suitable tests and test data for a give scenario

- **Additional Programming Techniques**
  - Use trace table to follow an algorithm containing selection and/or iteration.
  - Use of either count controlled iteration (for) or condition-controlled iteration (while / do until) to refine an existing algorithm
  - Use sub programs (functions and procedures) to produce structured code
  - The use of arrays (or equivalent) when solving problems, including both one-dimensional (1D) and two-dimensional arrays (2D)
  - The use of basic string manipulation
    - name = “Mr Smith”
    - Concatenate/Join strings *e.g. print greeting + name*
    - Length *e.g. print name.length or len(name)*
    - Substrings *e.g. name.substring(3,5)* will return Smith
    - Substrings e.g. name.left(2) will return Mr
    - Substrings e.g. name.right(5) will return Smith
    - Upper/Lower Case *e.g. name.upper or name.lower*
- **File Handling**
  - Opening, closing, writing and reading text files (refer to OCR spec GCSE (9-1) Computer Science J277 Specification (ocr.org.uk) page 32.

- **Data Types –** The use of different data types (integer, real, Boolean, character / string) and casting (converting between data types)
-

- **Integrated Development Environment (IDE)**
  - How Common tools/features available within an integrated development environment can help a programmer and how they can help:
    - Editors
    - Error Diagnostic
    - Debuggers
    - Variable Watch window
    - Step features
    - Breakpoints
    - Syntax Highlighting
    - Line numbers
    - Autocomplete
    - Auto indentation
    - Runtime environment and Inbuilt Translators
  - Characteristics and purpose of different levels of programming language (high and low level)
  - The purpose and need for translators (Compilers and Interpreters) and the benefit/drawback of each
  -
- **Maintainability of code** – How to write code that is easy to maintain (comments, clear indentation, sensible names for variables, sensible names for subroutines i.e. functions/procedures, using subroutines/subprograms such as functions and procedures

- **Defensive Design**
  - Input Validation: Understanding of the issues a programmer should consider ensuring that a program caters for all likely input values. Understanding of how to deal with invalid data in a program e.g. usernames and passwords.
  - Authentication to confirm the identity of a user e.g., Two Step Authentication

- **Searching Algorithms:**
  - Linear Search: Understand the main steps of a linear search algorithm.
  - Binary Search: Understand the main steps of a binary search algorithm.
  - Understand any pre-requisites/pre-requirements of an algorithm e.g. binary needs to be in order
  - Identify a search algorithm if given the code or pseudocode for it.
  - SQL: Be able to create SQL commands to search a database

- **Sorting Algorithms**
  - Bubble Sort: Understand the main steps of a bubble algorithm.
  - Merge Sort: Understand the main steps of a merge sort algorithm.
  - Insertion Sort: Understand the main steps of a insertion sort algorithm.
  - Identify a sort algorithm if given the code or pseudocode for it.

- **Boolean/Binary Logic**
  - Create Boolean/Binary Logic Diagrams from binary expressions e.g. draw the logic diagram for P = NOT A OR B
  - Create/Complete Truth tables from Boolean Logic expressions e.g. complete the truth table for P = NOT A OR B
  - Create/Complete Truth tables for Boolean Logic
  - Using and combining the operators AND, OR and NOT

- Applying logical operators in truth tables to solve problems e.g. create an algorithm from this Boolean logic diagrams

Revision Resources Paper 1:

- Teams Notebook – Keyword notebook and Task Notebook
- Programming constructs The three basic programming constructs
  - Sequence - Sequence - Programming constructs
  - Selection - Selection - Programming constructs
  - Count controlled iteration - Count-controlled iteration
  - Condition controlled iteration - Condition-controlled iteration
- Data Representation:
  - Images Representing images
  - Sound Representing sound
  - Units of Measurement (Bit, Byte, Kilobyte, Megabyte, Gigabyte, Terabyte)
  - Converting Binary to Denary https://www.youtube.com/watch?v=q7nZbAUTSC4
  - Converting Denary to Binary https://www.youtube.com/watch?v=70lM1qAD5u4
  - Binary Addition https://www.khanacademy.org/math/algebra-home/alg-intro-to-algebra/algebra-alternate-number-bases/v/binary-addition
  - Converting Hexadecimal to Denary Converting Hexadecimal to Denary - YouTube
  - Converting Denary to Hexadecimal Converting Denary to Hexadecimal - YouTube
  - Converting Hexadecimal to Binary Converting Hexadecimal to Binary - YouTube
  - Converting Binary to Hexadecimal Converting Hexadecimal to Binary - YouTube
  - Binary Logic
    - Types of logic gates
    - Combining Logic Gates
- Algorithms:
  - Sequence - Sequence - Programming constructs
  - Selection - Selection - Programming constructs
  - Count controlled iteration - Count-controlled iteration
  - Condition controlled iteration - Condition-controlled iteration
  - Creating algorithms/programs to solve problems.
  - Tracing algorithms/programs