

Year 12 Computer Science Mock Revision

This will be a 2 hour examination, using examination questions from the theory topics we have covered in lessons so far.

Topics to revise from Mr. Hall's lessons ...

Component 1 Section 5 - Systems Analysis

Investigation

- Describe the different methods of investigation.

Analysis

- Analyse a problem using appropriate techniques of abstraction and decomposition.
- Represent and interpret systems in an appropriate diagrammatic form showing the flow of data and the information processing requirements.
- Describe the selection of suitable software and hardware to address the requirements of a problem.

Changeover

- Describe the various methods of changeover: direct, pilot, phased and parallel, identify the most suitable method for a given situation and its relative merits.

Program Testing

- Describe the use of alpha, beta and acceptance testing.

Component 1 Section 6 - System Design

- HCI - Discuss contemporary approaches to the problem of communication with computers.

Component 2 Section 4 - Organisation and Structure of Data

- Define a file in terms of records and fields.
- Explain fixed and variable length fields and records and give examples of the appropriate use of each type.
- Distinguish between master and transaction files.
- Distinguish between the use of serial and sequential file access methods in computer applications.

Component 2 Section 5 - Databases

- Explain and apply entity relationship modelling and use it to analyse simple problems.
- Describe the use of primary keys, foreign keys, and indexes.
- Explain and apply appropriate techniques for data validation and verification of data in databases.
- Construct and run queries using Structured Query Language (SQL).

Topics to revise from Mr. Reid's lessons ...

Component 1 Section 1 – Data Structures

Describe, interpret and manipulate data structures including arrays (up to three dimensions), records, stacks, queues, trees and linked lists.

Describe the manipulation of records and arrays.

Represent the operation of stacks and queues using pointers and arrays.

Represent the operation of linked lists and trees using pointers and arrays.

Select, identify and justify appropriate data structures for given situations.

Component 1 Section 2 – Logical Operations

Draw truth tables for Boolean expressions consisting of AND, OR, NOT, XOR, NAND and NOR logical operations.

Apply logical operations to combinations of conditions in programming, including clearing registers, masking, and encryption.

Simplify Boolean expressions using Boolean identities and rules.

Component 1 Section 3 – Algorithms and Programs

Explain the term algorithm and describe common methods of defining algorithms, including pseudo-code and flowcharts.

Mathematical operations

Identify, explain and use mathematical operations in algorithms, including DIV and MOD.

Validation and verification

Identify, explain and apply appropriate techniques of validation and verification in algorithms and programs.

Sorting

Explain the need for a variety of sorting algorithms both recursive and non-recursive.

Describe the characteristics of sorting algorithms: bubble sort and insertion sort.

Explain the effect of storage space required, number of comparisons of data items, number of exchanges needed and number of passes through the data on the efficiency of a sorting algorithm.

Component 1 Section 4 – Principles of Programming

Explain the nature and relative advantages of different programming paradigms, and identify possible situations where they may be used.

Describe the distinguishing features of different types of programming paradigms, including procedural, event driven, visual and mark-up languages.

Describe the role of an object-oriented approach to programming and the relationship between object, class and method.

Describe the need for the standardisation of computer languages, and the potential difficulties involved in agreeing and implementing standards.

Component 2 Section 3 - Data Representation and Data Types

Representation of data as bit patterns

Explain the terms bit, byte and word.

Describe and use the binary number system and the hexadecimal notation as shorthand for binary number patterns.

Storage of Characters

Describe how characters and numbers are stored in binary form.

Describe standardised character sets.

Data types

Describe the different primitive data types: Boolean, character, string, integer and real.

Describe the storage requirements for each data type.

Representation of numbers as bit patterns

Apply binary arithmetic techniques.

Explain the representation of positive and negative integers in a fixed-length store using both two's complement, and sign and magnitude representation.

Describe the nature and uses of floating point form.

State the advantages and disadvantages of representing numbers in integer and floating point forms.

Convert between real number and floating point form.

Describe truncation and rounding, and explain their effect upon accuracy.

Describe the causes of overflow and underflow.