Exam 2: 1h 30 mins

2.1 Algorithms
1. Computational thinking
abstraction
decomposition
algorithmic thinking
2. Standard searching algorithms:
binary search
linear search
3. Standard sorting algorithms:
bubble sort
merge sort
insertion sort
4. How to produce algorithms using:
pseudocode
flow diagrams
5. Interpret, correct or complete algorithms.

2.2 Programming Techniques
1. The use of variables, constants, operators, inputs, outputs and assignments
2. The flow of a program:
sequence
selection
iteration(count and condition controlled loops)
3. The use of data types and casting:
Integer, real, Boolean, character and string
4. The common arithmetic operators + * / ** -
5. The common Boolean operators AND OR NOT
6. The use of basic string manipulation
7. The use of basic file handling operations:
Open, read, write, close
8. The use of records to store data
9. The use of SQL to search for data
10. The use of arrays (one and two dimensional)
11. How to use sub programs (functions and procedures)

2.3 Producing robust programs
1. Defensive design considerations:
input validation
planning for contingencies
anticipating misuse
authentication
2. Maintainability:
comments
indentation
variable names
3. The purpose of testing
4. Types of testing:
iterative
final / terminal
5. How to identify syntax and logic errors
6. Selecting and using suitable test data.

2.4 Computational logic
1. Why data is represented in computer systems in binary form
2. Simple logic diagrams using operators AND, OR and NOT
3. Truth tables
4. Combining Boolean operators using AND, OR and NOT to two levels

2.5 Translators and facilities of languages
1. Characteristics and purpose of different levels of programming language, including low level languages
2. The purpose of translators
3. The characteristics of an assembler, a compiler and an interpreter
4. Common tools and facilities available in an integrated development environment (IDE):
editors
error diagnostics
run-time environment
translators.

2.6 Data Representation
1.Units
bit, nibble, byte, kilobyte, gigabyte, terabyte, petabyte
2. Numbers
denary into binary vice versa
binary addition and overflow errors
binary shifts
binary / denary into hexadecimal and vice versa
check digits
3. Characters
the use of binary codes to represent characters
the term 'character set'
ASCII, extended ASCII and Unicode
4. 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
5. Sound
how sound can be sampled and stored in digital form
how factors affect the file size and the quality of playback:
sample size
bit rate
sampling frequency
6. Compression
need for compression
lossy/lossless compression