J277/02 Computational thinking, algorithms and programming – Component 2 exam

	☺	😐	☹	Revision materials created	Exam questions answered
2.1.1 Computational thinking					
Define the different 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					
Define Structure diagrams					
Create, interpret, correct, complete, and refine algorithms using: ○ Pseudocode ○ Flowcharts ○ Reference language/high-level programming language					
Identify common errors					
Trace tables – work through a trace table and explain the need for them.					
2.2.1 Programming fundamentals					
The use of variables, constants, operators, inputs, outputs and assignments					
The use of the three basic programming constructs used tocontrol the flow of a program: ○ Sequence ○ Selection ○ Iteration (count- and condition-controlled loops)					
Identify the common arithmetic operators: ○ MOD ○ DIV ○ Exponent					
The common Boolean operators AND, OR and NOT					
2.2.2 Data types					
I can explain the different datatypes: ○ Integer ○ Real ○ Boolean ○ Character and string ○ Casting					
2.2.3 Additional programming techniques					
The use of basic string manipulation – upper, lower, concatenation, etc.					
The use of basic file handling operations: ○ Open ○ Read ○ Write ○ Close					
The use of records to store data					
The use of SQL to search for data – SELECT, FROM, WHERE and *					
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					
How to use the Random number generation					
The relationship between the number of bits per character in acharacter set, and the number of characters which can be represented, e.g.: ○ ASCII ○ Unicode					

2.3.1 Defensive design					
Defensive design considerations: ○ Anticipating misuse ○ Authentication					
Explain Input validation					
Maintainability: ○ Use of sub programs ○ Naming conventions ○ Indentation ○ Commenting					
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					
Create a truth table based on a logic diagram					
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)					
Identify and explain the Common tools and facilities available in an Integrated Development Environment (IDE): ○ Editors ○ Error diagnostics ○ Run-time environment ○ Translators					