

COMPUTER SCIENCE

J277/02

COMPUTATIONAL THINKING,
ALGORITHMS AND
PROGRAMMING
REVISION CHECKLIST



CAMBRIDGE
OCR

2.1 ALGORITHMS

	Not Confident	Slightly Confident	Extremely Confident
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 PROGRAMMING FUNDAMENTALS

	Not Confident	Slightly Confident	Extremely Confident
2.2.2 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 and NOT			
2.2.2 Data Types			
The use of data types:			
Integer			
Real			
Boolean			
Character and string			
Casting			
2.2.3 Additional Programming Techniques			
The use of basic String manipulation			
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			
The use of arrays (or equivalent) when solving problems, including both one and two dimensional arrays			
How to use sub programs (functions and procedures) to produce structured code			
Random Number Generation			

2.3 PRODUCING ROBUST PROGRAMS

	Not Confident	Slightly Confident	Extremely Confident
2.3.1 Defensive design			
Defensive design considerations:			
Anticipating misuse			
Authentication			
Input validation			
Maintainability:			
Use of Sub Programs			
Naming Conventions			
Indentation			
Comments			
2.3.2 Testing			
The purpose of testing			
Types of testing:			
Iterative Testing			
Final/terminal Testing			
How to identify syntax and logic errors			
Selecting and using suitable test data.			
Normal Test Data			
Boundary Test Data			
Invalid/Erroneous Test Data			
Refining Algorithms			

2.4 BOOLEAN LOGIC

	Not Confident	Slightly Confident	Extremely Confident
2.4.1 Boolean logic			
Simple logic diagrams using the operations AND, OR and NOT			
Truth tables			
Combining Boolean operators using AND, OR and NOT			
Applying logical operators in appropriate truth tables to solve problems			

2.5 PROGRAMMING LANGUAGES AND INTEGRATED DEVELOPMENT ENVIRONMENTS

	Not Confident	Slightly Confident	Extremely Confident
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 an assembler, a compiler and an interpreter			
2.5.2 The Integrated Development Environment (IDE)			
Common tools and facilities available in an integrated development environment (IDE):			
Editors			
Error diagnostics			
Run-time environment			
Translators.			