GCSE Computing (J276): Personal Progress Record Component 02:Computational thinking, algorithms and programming

2.1 Algorithms	RED	AMBER	GREEN
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 programs.			
2.2 Programming Techniques			
1. The use of variables, constants, operators, inputs, outputs and assignments			
2. The use of the three basic programming constructs used to control the flow of a program:			
sequence			
selection			
iteration(count and condition controlled loops)			
3. The use of basic string manipulation			
4. The use of basic file handling operations:			

open			
read			
write			
close			
5. The use of records to store data			
6. The use of SQL to search for data			
7. The use of arrays (or equivalent) when solving problems, including both one and two dimensional arrays			
8. How to use sub programs (functions and procedures) to produce struictured code)			
9. The use of data types:			
interger			
real			
Boolean			
character and string			
casting			
10. The common arithmetic operators			
11. The common Boolean operators.			
2.3 Producing robust programs			
1. Defensive design considerations:			
input sanitisation / validation			
planning for contingencies			
anticipating misuse			
authentication			
2. Maintainability:			
comments			
indentation			
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			
5. Applying logical operators in appropriate truth tables to solve problems			
6. Applying computing-related mathematics:			
+			
-			
/			
*****			
exponentiation (^)			
MOD			
DIV			
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			
how data needs to converted into a binary format to be processed by a computer			
2. Numbers			
how to convert positive denary whole numbers (0-255) into 8 bit binary numbers and vice versa			
how to add two 8 bit binary integers and explain overflow errors which may occur			
binary shifts			
how to convert positive denary whole numbers (0-255) into 2 digit hexadecimal numbers and vice cersa			
how to convert from binary to hexadecimal equivalents and vice versa			
check digits			
3. Characters			

the use of binary codes to represent characters			
the term 'character set'			
the relation ship between the number of bits per character set abd the number of characters which can be represented (for example 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 sampling intervals and other factors affect the size of a sound file and the quality of its playback			
sample size			
bit rate			
sampling frequency			
6. Compression			
need for compression			
types of compression			
lossy			
lossless			