Computing Scheme of Assessment 2018

Grade	Generic ICT Skills	Computational Thinking	Visual Programming	Textual Programming	Binary & Data Representation
9.5				Subprograms, programming constructs, data validation and the choice of data types and structures lead to an overall program which is fully functional.	
8.5		Full decomposition into subprograms has been made clear abstraction.		A full solution has been built with little logic errors, showing the effective use of debugging skills.	
7.5		Full decomposition into sub programs has been made with minor errors so does not lead to clear abstractions.		Subprograms, programming constructs, data validation & the choice of data types and structures lead to an overall program which is functional with minor omissions.	
6.5		An attempt at decomposition into subprograms has been made.		The program makes some use of subprograms, programming constructs, data validation, with suitable choice of data types.	Understand how computers represent and manipulate positive and negative numbers (sign & magnitude, and two's complement).
5.5		Analyse different algorithms for the same problem and use logical reasoning to detect and correct errors and improve the algorithm.		The program addresses the requirements of a problem, but may include major omissions, there are some significant syntax and logic errors present.	Be able to calculate the file size of a sound file.
4.5		Think through an algorithm and predict an output.		Create a complex program that incorporates constructs, data validation and appropriate choice of data types.	Understand why hexadecimal notation is used.
3.5	Develop capability, creativity and knowledge in Computer Science, Digital Media and Information Technology.	To understand and explain abstraction.	Solve problems by decomposing them into smaller parts in a visual language.	Be able to insert comments explaining what the program does.	Identify the three main logic gates.
2.5	Take into account others comments and analyse usefulness.	To understand and explain decomposition.	Debug programs that accomplish specific goals in a visual language.	Create a more complex program that makes use of loops, variables, arrays.	Understand how to convert between the terms: bit, nibble, byte, kilobyte (KB), megabyte (MB), gigabyte (GB) andterabyte (TB).
1.5	Use keywords in a search engine to select information in an efficient manner.	Be able to perform pattern recognition.	Debug simple programs in a visual language.	Create a program that correctly declares one or more variables.	Perform calculations of 2-bit binary numbers.
0.5	Save files using names that help to find them again.	Describe the term pattern recognition.	Use various forms of input & output in a visual language.	Be able to read a simple set of instructions and predict what the outcome will be.	Identify how binary and denary numbers differ