Computer Science Scale and Criteria

Steps	Computational Thinking	Theoretical Knowledge	Digital Literacy
Team 1	Can use the terms search and sort.	Can identify different parts of a computer system but may not be able to describe their purpose.	Can use a keyboard and mouse to enter data
Team 2	Can identify an input/output	Define terms hardware and software	Can use a variety of input methods to achieve the same outcome on a computer (e.g copy and paste)
Team 3	Can describe an input/output	Identify a computer network Identify and give examples of system software and application software	Can take a screenshot of completed work.
1	Can create a code that allows for the output and input of data. Can identify different data types	Understand denary Understand what a pixel is Understand relationship between hardware and software	Can use PowerPoint or Word to make a basic document which may not read as intended
2	Can understand and explain the term algorithm	Describe binary	Can use PowerPoint and Word to make a basic

	Understands and uses Arithmetic operations in a programming language Understands and applies the following programming concepts: 1. Variables 2. assignment	Describe a bitmap in terms of size of pixels and colour depth Explain that sound is analogue and computers are digital, one must converted to the other. Describe main types of computer networks	document which reads as intended
3	Can understand and explain the terms abstraction and decomposition. Can identify steps to solve a problem Understands the need for meaningful identifier names Understands and uses relational operations in a programming language Define most key terms associated with the classification of programming languages	Can convert between binary and denary Explain and describe character encoding Describe functions of OS and utility software Describe how number of pixels and colour depth will effect size of bitmap image Can explain star and bus network topologies Construct a truth table for OR, AND, NOT Define some key terms of cyber security	Can use PowerPoint or Word to make a coherent document, can create a basic spread sheet being able to enter information and attempt formulas
4	Can use a flow chart to represent and algorithm	Can describe Hexidecimal	Understands how to use the internet safely when

	Understands and uses Boolean operations in a programming language
	Understands and applies the following programming concepts
	    1. selection
	    2. constants, iteration,
		Can convert between binary and hexadecimal
		Can explain why hexadecimal is often used.
		Describe digital representation of sound in terms of sampling rate and sample resolution
		Can construct a truth table and logic circuit diagram with up to three inputs
		Can define and explain the purpose of common network protocols
		Define the term 'embedded system'
		Can explain some different types of memory and storage including secondary storage
			searching for information, can apply formulas to data handling software accurately and can test their work to ensure that it works.

5	Can explain and use pseudocode to represent a simple algorithm	Can convert between hexadecimal and denary Can apply binary addition	Can explain in detail what software is being used for a task based on the user's requirements, is able to use validation techniques when

	Can use definite and indefinite iterations (for, while and repeat) Understands and applies the concept of a subroutine. Design a solution that obtains input/displays output Can use nested selection explain the differences between program translators	Calculate bitmap file size based on number of pixels and colour depth Explain different areas that OS manages Explain Von Neumann architecture Explain the current ethical, legal and environmental impacts and risks of digital technology on society Can most different types of memory and storage including secondary storage	using data handling software
6	Can understand and explain how different sort and search algorithms work. Can use nested selection and iteration Can use a random number generator Define all key terms associated with the classification of programming languages	Can explain and apply binary shift Convert binary data into b/w image and vice versa Explain what data compression is and why it may be compressed in more than one way (middle out?) Can fully explain the role, operations and requirements of the CPU	Can research a user's needs before making a decision about how to produce a required product – the results of which will be evident in designs and final products, can also apply complex data handling tools (for example advanced queries) and represent this in a report

		Define most key terms of cyber security Can all different types of memory and storage including secondary storage	
7	Can compare and explain the efficiency of different algorithms (including search and sort algorithms) Can use a 1 dimensional array in the design of a solution Understand and able to use string handling operations in a programming language	Calculate sound file size based on sampling rate and sample resolution Can explain the Fetch-Execute Cycle Can explain advantages/disadvantages of cloud storage vs local storage Define all key terms of cyber security	Can evaluate a user's needs and represent these needs in multiple designs before applying one design to a successful product and can apply complex validation in data handling software to ensure all data is accurate
8	Can use a 2 dimensional array in the design of a solution Compare advantages and disadvantages of low level vs high level programming as well as when each type of translator would be appropriate	Explain Huffman coding and interpret Huffman Trees, calculate number of bits required to store data compressed using HC or uncompressed data stored ASCII Explain how an embedded system differs from a	Can create a clear success criteria for the users' needs, can evaluate a range of designs for suitability, will resolve all errors and any improvements given by the user, will clearly identify how successful the work has been based on the clear

		non-embedded system. Understand and explain methods of network security	success criteria. Will also be able to use macros in data handling software to make jobs more efficient
9	Has a full understanding of subroutines (see GCSE spec) Can write simple authentication and data validation routines Design a solution that will read/write text to a file.	Explain how data can be compressed using run length encoding and represent data in RLE frequency/data pairs Fully understand and explain the 4 layer TCP/IP model Explain aspects of Social Engineering and Methods to Detect and Prevent Security Threats.	Can explain what software is most appropriate when completing a given task but is able to use a range of software to produce the most professional end product possible. Can evaluate their product and will suggest further improvements and future designs that could be produced based on the evaluation. Can use data handling software to create advanced solutions to complex problems