Curriculum Map - KS4 Computer Science (J277)

Subject: Computer Science **Year Group:** Year 10

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Content *Descriptive/propositional knowledge* *'knowing that'*	E-safety revision 2.1 Algorithms Programming understand and apply the fundamental principles and concepts of Computer Science, including abstraction, decomposition, logic, algorithms, and data representation	2.2 Programming fundamentals Programming know that effective programs require the use of programming constructs analyse problems in computational terms through practical experience of solving such problems, including designing, writing and debugging programs	2.3 Producing robust programs 2.4 Computational logic 3.0 Programming Project (3 wks) know that each logic gate has a corresponding truth table	3.0 Programming Project (3 wks) 2.5 Programming languages and IDEs 2.6 Data representation think creatively, innovatively, analytically, logically and critically	3.1 Programming Project (2 wks) Mock preparation know that problems can be solved using computational thinking and know how to apply it through a chosen programming language.	Mocks / mock feedback programming project /theory contingency know that the exam covers computer systems,programming, computational thinking, and algorithms
Skills *Ability knowledge* *'knowing how'*	know how to use computational methods such as abstraction, decomposition and algorithmic thinking when solving problems know the standard search and sort algorithms and be able to identify them if given the code.	know how to use variables, constants, operators, inputs, outputs and assignments Know how to use the three basic programming constructs to control the flow of a program Know how to use basic string manipulation and basic file handling	know how to use skills from Component 01 and Component 02 to create a programmable solution to a set problem. know how and why defensive design methods are used in programming Know how to implement maintainability in a	know the characteristics and purpose of different levels of programming language know the differences between high- and low-level programming languages know the purpose of translators and the characteristics of a	know how to accurately answer the various types of examination question and apply the correct knowledge know how to plan for and answer long answer questions	know how to self-assess individual progress according to feedback given by the teacher know how to reACT to feedback given by the teacher and improve knowledge where needed

	know how to produce algorithms using pseudocode and flow diagrams and can interpret correct or complete algorithms.	operations: open, read, write, close know about the use of records to store data and the use of SQL to search for data know about the use of arrays (or equivalent) when solving problems, including both one and two dimensional arrays how to use subprograms (functions and procedures) to produce structured code	program and justify its use. know the purpose of testing and the various means of testing and be able to select and use suitable test data. Know how to identify syntax and logic errors know why data is represented in computer systems in binary form know how to create simple logic diagrams using the operations AND, OR and NOT know how to create truth tables, combine Boolean operators to two levels know how to use truth tables to solve problems know about applying computing-related mathematics inc. exponentiation, MOD and DIV Know how data is represented and the reasons compression is used	compiler and an interpreter know the common tools and facilities available in an Integrated Development Environment (IDE)		

Key Questions	What is computational thinking? What is pseudocode and why do we use it? How does it differ from a regular programming language?	How is data stored and accessed when a program is executed? What is SQL? Why and how is it used? Why do we need to use subprograms?	What is a robust program? What is input validation? Why is it important to include it when coding? How many marks is the project worth How long is allocated for the project?		What are the units that will be included in the mock? How many marks is the paper out of? How long is the mock examination?	What are my areas of strength? Which areas have I shown to have gaps in knowledge? Which type of question did I find most difficult?
Assessment (Each topic is marked out of 20)	3 week assessment, End of term test	3 week assessment, End of term test	3 week assessment, End of term test	3 week assessment, End of term test	Practice questions , past papers and feedback	Mock Exam
Literacy/ Numeracy/ SMSC/ Character	Development in communication/ literacy skills/apply mathematical skills relevant to Computer Science.	Development in communication/ literacy skills/apply mathematical skills relevant to Computer Science.	Development in computing-related mathematics literacy skills specifically report writing and technical report writing	Development in communication/ literacy skills/apply mathematical skills relevant to Computer Science.	Development in communication/ literacy skills specifically report writing and technical report writing	Development in communication/ literacy skills
Futures	Emphasis on the mathematical skills used to express computational laws and processes. This qualification is suitable for learners intending to pursue any career in which an understanding of technology is needed. Students typically go on to A Levels in Computer Science or IT.					
Enrichment	Cyber Discovery Challenge (new challenge every year). https://joincyberdiscovery.com/	Cyber Discovery Challenge	Cyber Discovery Challenge	Cyber Discovery Challenge	Cyber Discovery Challenge	