Curriculum Map
Year 12

Computer Science

Learning Aims and Curriculum Intent:

The A-level specifications in computer science encourages students to develop: an understanding of, and the ability to apply, the fundamental principles and concepts of computer science, including abstraction, decomposition, logic, algorithms, and data representation. The ability to analyse problems in computational terms through practical experience of solving such problems, including writing programs to do so. The capacity for thinking creatively, innovatively, analytically, logically, and critically. The capacity to see relationships between different aspects of computer science. Mathematical skills related to: Boolean algebra, comparison and complexity of algorithms, number representations and bases. The ability to articulate the individual (moral), social (ethical), legal and cultural opportunities and risks of digital technology.

Term	Content, Key Questions and Knowledge	Skills	Assessment
Michaelmas	**4.1 Fundamentals of programming** What are the building blocks of a program? Explain how exception handling can be used to make source code more robust. To be able to apply and synthesise a range of programming techniques and paradigms to create a solution to a provided problem: Specification Link **4.2 Fundamentals of Data Structures** How do I represent that in a program? Learn about a series of abstract data types, lists, stacks, queues, vectors, hash tables. Specification Link **4.3 Fundamentals of algorithms** What are common ways to solve problems? Differentiate between the mechanisms of a breadth-first and depth-first graph traversal. Be able to describe and trace the execution of common algorithms used in Computer Science. Specification Link **4.7 Fundamentals of computer organisation and architecture** What goes on inside my computer? Have an understanding and knowledge of the basic internal components of a computer system. Specification link. **4.10 Fundamentals of databases** How can I organise my data? This covers entity relationship diagrams, normalisation and an introduction to SQL. Students will get the opportunity to program a system using python and sqlite. Specification Link	Know the most common procedural programming constructs. Understand how and where abstract data structures are used in programming. Know the common algorithms and be able to model them. Understand the internal components of a computer, how they relate to each other, how they work and how they can be programmed using low level languages. Understand how data is organised in order to make it useful and understandable.	Retrieval practice starters throughout the course. End of topic tests. Regular quizzes using Smart Revise and Isaac Computer Science.
Lent	**4.4 Theory of computation** What is the theory of solving problems? Determine the type of abstraction being applied in [a given] scenario. Be able to describe and apply the methods of abstraction in order to derive a computational model: Specification Link **4.12 Fundamentals of functional programming** How can I manipulate my data without changing it? This unit is an exploration of a different programming paradigm, students get the opportunity to program using Haskell. Specification Link **4.9 Fundamentals of communication and networking** How does the internet work? Establish a possible IP address for a provided subnet on a network diagram. Be able to describe the common hardware, software and techniques used in transmitting data over a network. Specification Link	Be able to write and follow algorithms using different techniques. Learn how to write functional programs using Haskell. Understand the hardware and software involved in all types of networking.	Retrieval practice starters throughout the course. End of topic tests. Regular quizzes using Smart Revise and Isaac Computer Science.

Trinity	**4.6 Fundamentals of computer systems** Is it true or false? Simply the provided Boolean expression. Be able to describe the fundamental characteristics of hardware and software, as well as apply simplification techniques to Boolean expressions. Specification Link **4.11 Big Data** How do I use all the data I have? An introduction to the properties and uses of Big Data, that uses concepts introduced in unit 4.12 functional programming, and flows into 4.8 consequences of the uses of computing. Specification Link **4.5 Fundamentals of data representation** Determine the set(s) of numbers to which a provided value belongs, applying common set operations as required. To be able to identify common mathematical number sets and to describe their significance in Computer Science. Specification Link **4.8 Consequences of uses of computing** Explain how the Data Protection Act affects how social media companies might operate. To be able to define common pieces of computer legislation and evaluate their impact on modern computing conventions. Specification Link	Be able to solve Boolean algebra problems. Understand the concept of Big Data and recognise how it may be used. Know how different types of data are represented using binary. Understand the responsibilities to society of developments in the technology field.	Retrieval practice starters throughout the course. End of topic tests. Regular quizzes using Smart Revise and Isaac Computer Science.

l

What consolidation looks like in this subject	Application of GCSE programming skills to new and more challenging situations. Development of programming skills to include advanced concepts such as OOP and low-level programming. Developing understanding of theoretical concepts to A-Level standard (eg OSI model)	
Examples of Homework	Worksheets reinforcing lesson content. Problem solving challenges. Practice examination style questions.	
Key terminology	AQA | Subject specific vocabulary	
Super-curricular enrichment and scholarly extension	• **Read: Computing SharePoint Site (News Feed), MCI Resources** • **Watch: BBC Click,** • **Listen:** BBC Sounds - Podcasts (Technology Section) • **Visit:** 3D Virtual Tour — The National Museum of Computing (tnmoc.org)	
Useful websites	https://isaaccomputerscience.org https://student.craigndave.org/ https://www.codewars.com/	
Who can I contact?	**Head of Department**	Mark Jalowiecki - msj@forest.org.uk
	Teachers	Vinh Kieu - vk@forest.org.uk