The Fisher Way: Curriculum

The Fisher Way aims to educate and inspire with joy, faith and love because we are an inclusive Catholic community.

Successful and resilient learners who aspire to and achieve excellence

Confident individuals who can explore and communicate effectively

Responsible citizens who are active, loving and wise in all their endeavours

Subject	Computer Science
Year Group	Year 10
Intent	**Successful and Resilient Learners:** who can analyse problems and find efficient, creative solutions for both real world situations and artificial systems **Confident Individuals**: who can use digital technology effectively, creatively and safely in their personal lives and future careers **Responsible Citizens**: who understand the social and cultural impact that technology has on their lives and the lives of others.

Narrative	The Year 10 Computer Science curriculum follows the Edexcel GCSE Computer Science specification and builds the knowledge pupils need for both papers: Paper 1, Principles of Computer Science, and Paper 2, Application of Computational Thinking, the on screen programming exam in Python. The autumn term focuses on theory. Pupils learn how data is represented in binary, including numbers, text, images and sound, and how files are compressed. They then study computer systems: the CPU and the fetch decode execute cycle, memory and storage, operating systems and utility software, embedded systems, and how programming languages are translated. In the spring term, pupils study networks, including protocols, topologies and network security, then the impacts and issues of computing, covering environmental, ethical and legal concerns and cyber security. They finish the term with computational thinking: decomposition, abstraction, algorithms, searching and sorting, logic and trace tables. The summer term is given to programming in Python. Pupils build secure foundations in variables, data types and selection, writing, testing and debugging their own programs in preparation for the on screen exam. By the end of Year 10, pupils will have covered most of the Paper 1 theory and be ready to develop their programming further in Year 11.					
Half term	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Knowledge (topics studied)	Theory: Data Theory: Computer Systems		Theory: Networks Theory: Impacts and Issues Theory: Computational Thinking		Programming: Variables, Data Types and Selection	
Key skills	Learners convert between binary, denary and hexadecimal, represent signed integers using two's complement, add binary numbers, identify overflow, and apply logical and arithmetic shifts. They explain how text, bitmap images and sound are represented, calculate file sizes, and compare lossless and lossy compression. They then describe the von Neumann architecture and the fetch decode execute cycle, compare RAM, ROM and secondary storage, explain the role of the operating system and utility software, and describe how compilers, interpreters and assemblers translate code.		Learners compare LANs and WANs, wired and wireless connections, and star and mesh topologies. They explain common protocols and the TCP/IP model, calculate data transmission times, and describe threats to networks and how they are protected. They then evaluate the environmental, ethical and legal impact of computing, including privacy, artificial intelligence and software licensing, and explain cyber security threats and defences. Finally, they apply decomposition and abstraction, interpret and write algorithms, use trace tables, compare linear and binary search with bubble and merge sort, and complete truth tables for AND, OR and NOT.		Learners write Python programs that take input, process it and produce output. They use variables and constants, choose appropriate data types, convert between them, and use arithmetic, relational and logical operators. They use if, elif and else to control the flow of a program, including nested selection, and test and debug their code to find and fix syntax and logic errors.	

Cultural capital	Learners discover how every photo, song and message they share is stored as binary, and why compression makes streaming and social media possible. They gain insight into what happens inside the devices they use every day, from phones to the embedded systems in cars and household appliances.	Learners understand how the internet and the networks behind it connect the world, and the risks that come with that connection. They consider the real impact of technology on society and the environment, from e waste to AI decision making, and learn the problem solving approaches used by software engineers.	Learners experience programming as it is done in industry, writing text based code to solve real problems. They develop the logical thinking, precision and persistence that programming demands, and see how these skills lead to careers in software development, engineering and data science.
Assessment	End of Unit Assessment (Data) End of Unit Assessment (Computer Systems)	End of Unit Assessment (Networks) End of Unit Assessment (Impacts and Issues) End of Unit Assessment (Computational Thinking) Mock Exams	Programming Tasks End of Unit Assessment