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 11
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 11 Computer Science curriculum completes the Edexcel GCSE Computer Science specification and prepares pupils for Paper 1, Principles of Computer Science, and Paper 2, Application of Computational Thinking, the on screen programming exam in Python. Theory and programming are taught side by side throughout the year. In the autumn term, pupils revisit and deepen their knowledge of data representation, the impacts and issues of computing, and networks. Alongside this, they secure the core programming structures of sequence, selection and iteration, using them to solve increasingly complex problems. In the spring term, pupils study computational thinking in depth: algorithms, searching and sorting, logic and trace tables. In programming, they use arrays to store and process data, write subroutines with parameters and return values, and read from and write to files. In Summer 1, pupils bring everything together through composite programming challenges that mirror the style of the on screen exam. By the end of Year 11, pupils will have covered the full specification and be ready to sit both exams.					
Half term	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Knowledge (topics studied)	Theory: Data Theory: Impacts and Issues Theory: Networks Programming: Programming Structures		Theory: Computational Thinking Programming: Arrays, Subroutines and Files		Programming: Composite Programming Challenges	
Key skills	Learners convert between number bases, use two's complement, binary addition and shifts, and calculate file sizes for images and sound. They evaluate the environmental, ethical and legal impact of computing and explain cyber security threats and defences. They explain network types, topologies, protocols and the TCP/IP model, and how networks are secured. In programming, they combine sequence, selection and iteration, using count controlled and condition controlled loops, input validation and string handling to solve multi step problems.		Learners apply decomposition and abstraction, interpret and complete algorithms in flowcharts and code, use trace tables, compare searching and sorting algorithms, and complete truth tables. In programming, they use one and two dimensional arrays to store and process data, break programs into subroutines with parameters and return values, use local and global variables appropriately, and read from and write to text files.		Learners solve complete exam style problems that combine every programming skill from the course, including data structures, subroutines, validation and file handling. They practise reading and amending existing code, writing programs from scratch to a time limit, and testing their	

			solutions against a specification.	
Cultural capital	Learners see how data, networks and the wider impacts of technology connect, and are able to discuss them with confidence and balance. Writing larger programs builds the resilience and problem solving that employers in every technical field look for.	Learners understand how professional programmers structure large programs so that they can be tested, reused and shared in a team. Working with stored data shows them how real software, from games to banking systems, keeps information between sessions.	Learners build the confidence to tackle unfamiliar problems independently and under time pressure, a skill that will serve them in further study and the workplace. This prepares them for A Level and vocational courses in computing and for careers in technology.	
Assessment	End of Unit Assessments Mock Exams	End of Unit Assessments Mock Exams	Paper 1 and Paper 2 Exams	