Yardleys Curriculum Aims

- To achieve academic excellence
- To educate the 'whole child' so they are ready for life
- To work collaboratively and ethically to provide education of the highest standard

COMPUTER SCIENCE – KEY STAGE 4

Curriculum Overview
INTENT: Through our ambitious computing curriculum our learners will have access to a broad and balanced range of topics including digital literacy, computational thinking and modern technology. Yardley's computing curriculum will provide students with an understanding of how computing underpins today's modern lifestyle and has made the world better, faster and more connected. We ensure that the students at Yardleys can develop to become masters and creators in this field, to aid them in their development of our rapidly changing technological world.
Year 10
In Year 10 students dive into the fundamental principles that power modern technology using programming as the backbone of our learning. Through hands on coding and problem solving, skills and knowledge to understand, design and analyse systems will be gained. Alongside this, students will start their journey into the solid foundations in the theory of Computer Science. Students will uncover how computers work at their core; next we will investigate how computers store and manage data and then how computers communicate. We build upon the year 9 unit of cybersecurity by looking at other threats and how systems defend against them. System software is what allows hardware and software to work together, this is where we look at the role of the operating system. Technology doesn't exist alone; students will explore the broader impacts of computing including privacy and legislation.

	Architecture Memory and Storage Programming	**Computer Networks Programming**	**Systems Software Issues and Programming**
SUBSTANTIVE KNOWLEDGE	• Primary storage • Secondary storage • Units • Data storage • Compression	• Types of networks • Network Hardware • Modes of connection • Protocols • Forms of attack/Prevention methods	• Operating systems • Utility software • Impacts of digital technology • Legislation

DISCIPLINARY KNOWLEDGE	• Why computers have memory and storage • Why is data stored as binary • How data storage is calculated • How to convert between the number forms • Understand and carry out binary shift • Understand how characters, images and sound are stored on a computer • Identify the advantages and disadvantages of lossy and lossless compression	• Understanding of different factors that can affect the performance of a network • The tasks performed hardware • The concept of the Internet and clients • Role of the DNS • Apply understanding of networks to a given scenario • Principles of protocols and encryption • How the attack is used & the purpose of the attack • What each prevention method may limit/prevent & how it limits the attack	• Purpose and functionality of OS and utility software • Understand the impacts of computer science issues • Purpose and suitability of each legislation • Features of open and proprietary software

Year 11

In Year 11 we dive deep into the art and science of problem solving, algorithms and programming – the heart of computational thinking. Year 11 is about mastering the skills to design, analyse and implement efficient solutions to real world challenges. Building on programming foundations from year 10, this year focuses on taking skills to the next level ensuring students can refine their ability to write efficient, modular and maintainable code using advanced programming techniques such as procedures, functions, string manipulation, file handing and error handling. Boolean logic is essential for controlling how computers make decisions, this unit will show how logic gates and truth table are constructed.

	Algorithms, Programming Techniques Robust Programs, IDE	**Boolean Logic Revision**	[redacted]
SUBSTANTIVE KNOWLEDGE	• Programming Constructs – Sequence, Selection & Iteration • Variables & Data Types - integer, real, Boolean, character, string • Type casting • Input and output • If statements • String manipulation	• Simple logic diagrams using the operators AND, OR and NOT • Truth tables • Combining Boolean operators using AND, OR and NOT Applying logical operators in truth tables to solve problems	
DISCIPLINARY KNOWLEDGE	• Understand and use data types: integer, real, Boolean, character and string • Declare and use constants and variables • Use input, output and assignment statements • Use random number generation • Write algorithms in pseudocode involving sequences • Use arithmetic operators including MOD and DIV	• Construct truth tables for the following logic gates: 1. NOT 2. AND 3. OR • Construct truth tables for simple logic circuits • Create, modify and interpret simple logic circuit diagrams	

	• Use string handling and conversion functions Use selection and nested selection statements • Use NOT, AND and OR when creating Boolean expression • Understand and use iteration in an algorithm • Write algorithms in pseudocode involving sequence, selection and iteration		[redacted]