YEAR 9 COMPUTER SCIENCE

PREPARING STUDENTS FOR TOMORROW, BIT BY BIT

The Computing department will help to create, share, and apply knowledge in all branches of Computer Science and ICT. We will educate students to be successful, ethical, and effective problem-solvers with a passion to innovate and create, rather than just passive consumers and users of technology. We will develop an understanding and appreciation of all aspects of digital products, from how they work to how they look. We will foster curiosity and encourage exploration to create students who can contribute positively to the well-being of our society and who are prepared to tackle the complex 21st Century challenges facing the world.

Summary focus areas:

- Innovate, create, develop
- Solving 21st Century problems
- Active developers not passive consumers

AUTUMN		SPRING		SUMMER
Logical thinking and reasoning, Boolean logic Binary and hexadecimal	Data representation Compression	Algorithms Searching and sorting	Pseudo-code	Java programming

Homework for Computing is set weekly to support and extend the students' studies from their lessons. Work may be a mixture of practical, computer-based tasks and paper-based written work or design tasks. Activities set as homework may be:

- Preparatory work or research ahead of a new topic or concept being discussed in lessons.
- Extension work that allows the student to explore a topic in more depth or in other contexts.
- Application work that allows students to practise skills or demonstrate abilities.

Students are expected to spend around an hour on a homework activity each week and work is marked promptly to help students to identify and understand their weaknesses to make incremental improvements over the course of the year.

UNIT	DURATION (LESSONS)	LEARNING OBJECTIVES/OUTCOMES
Logical thinking and reasoning Boolean logic	10	• Understand the concept of logic • Understand the basic logical operations – *and, or, not* • Understand how to approach problems in a logical, structure and thorough manner • Be able to demonstrate logical thinking and reasoning through the completion of puzzles and logic problems • Be able to interpret and create logic circuit diagrams • Be able to construct truth tables for given logical contstructs
Binary and hexadecimal	6	• Understand what binary is and why it is used in computing • Be able to count in binary and convert numbers between counting binary and denary • Understand the uses of hexadecimal • Be able to convert between hexadecimal, binary and denary

Data representation	10	• Understand how binary can be used to represent text, images and sound
		• Explore binary representation systems and concepts such as:
		  ○ ASCII
		  ○ Unicode
		  ○ Bitmap images
		  ○ Colour depth and resolution
		  ○ Sound sampling, rates and resolutions

Compression	6	• Understand the measurement of data storage • Explore the impact of file size on real-life applications • Understand how data compression can have positive and negative effects • Be able to apply the Run Length Encoding and Huffman Coding compression techniques
Algorithms	10	• Understand that algorithms are computational solutions that always finish and return an answer • Understand the concepts of decomposition and abstraction in helping to develop algorithms • Be able to interpret simple algorithms to deduce their function • Be able to create algorithms to solve simple problems • Be able to detect and correct errors in simple algorithms • Be able to evaluate the efficiency of algorithms

Searching and sorting	8	• Be able to explain the uses and importance of searching and sorting procedures • Understand the algorithms for Bubble Sort and Merge Sort • Be able to apply sorting algorithms to given data sets and evaluate their efficiency • Understand the algorithms for Linear Search and Binary Search • Be able to apply searching algorithms to given data sets and evaluate their efficiency
Pseudo-code	6	• Be able to express simple algorithms in pseudo-code • Be able to understand and interpret pseudo-code algorithms to determine their function • Be able to complete trace tables for given algorithms to determine the likely output

Java programming	20	• Understand what is meant by the terms data and information
		• Be able to describe the difference between a constant and a variable
		• Understand when to use constants and variables in problem solving scenarios
		• Understand the different data types available to them, specifically:
		  ○ integer
		  ○ boolean
		  ○ real
		  ○ character
		  ○ string
		• Be able to explain the purpose of data types within code
		• Be able to program with 1 and 2 dimensional arrays
		• Be able to use selection and iteration to control program flow and understand the uses of different type of loop
		• Understand the need for structure when designing coded solutions to problems
		• Be able to solve simple problems using code