

Subject: Computing

Homework: Set on Satchel One every two weeks.

Year group	Term	Curriculum Focus
7	Autumn 1	Students learn how to use the school network safely, organise their files clearly, and communicate professionally by email. They practise creating simple presentations and develop confidence using digital tools.
7	Autumn 2	Students learn how to stay safe online, create strong passwords and recognise suspicious messages. They explore malware, phishing and how to report concerns safely. The focus is on building responsible digital citizens.
7	Spring 1	Students learn what the main parts of a computer do, including the CPU, RAM and storage devices. They explore input and output devices and design a simple virtual PC.
7	Spring 2	Students learn how to control sprites using sequencing, selection and repetition, building confidence in creating interactive projects. They develop problem-solving skills through debugging and refining their work.
7	Summer 1	In Summer 1, students are introduced to text-based programming using Python. They learn how to write simple programs using inputs, outputs and variables, and begin to apply computational thinking to solve problems.
7	Summer 2	In Summer 2, students explore how computers represent and store data. They are introduced to binary and learn how images and sound are stored digitally. Alongside this, pupils deepen their understanding of online safety.
8	Autumn 1	Students explore how computers store numbers, text, images and sound. They learn about binary, pixels and file sizes, and create simple digital artefacts to apply this knowledge.
8	Autumn 2	Students build on their coding skills by learning how computers make decisions and repeat actions. They practise writing programs using if-statements and loops, and learn how to test their code thoroughly.
8	Spring 1	Students learn how to use spreadsheets to calculate, organise and present data. They use formulas, functions and charts to model simple scenarios and make decisions.
8	Spring 2	Students strengthen their use of formulas, functions and data formatting to create clear, accurate spreadsheets. They test different scenarios to understand how data models support decision-making.
8	Summer 1	During Summer 1, students build upon their programming knowledge by learning how to use iteration (loops) and more advanced logic within Python. They begin creating larger, structured programs and develop greater independence in debugging and refining their work.
8	Summer 2	In Summer 2, pupils study core computer system concepts, including how the CPU works and the fetch–decode–execute cycle. They learn about different types of memory and storage and explore how networks enable computers to communicate.
9	Autumn 1	Students are introduced to programming through simple Python tasks. They learn how to use inputs, outputs, variables and basic calculations, and how to plan algorithms before coding. They begin to write small programs and develop early debugging skills.
9	Autumn 2	Students deepen their understanding of decision-making and repetition in Python. They learn how to combine conditions, use loops effectively and test programs using planned test cases.

Year group	Term	Curriculum Focus
9	Spring 1	Students learn how to structure programs using functions, lists and string processing. They design small modular applications and practise writing clear, reusable code.
9	Spring 2	Students deepen their understanding of functions, parameters and return values while building more structured programs. They learn to process strings and manage lists to store and manipulate data effectively.
9	Summer 1	In Summer 1, students further develop their programming skills by working with data structures such as lists and improving the efficiency and organisation of their code.
9	Summer 2	In Summer 2, students begin to study data representation, cybersecurity threats and prevention methods, and consider the ethical, legal and environmental impacts of technology.
10	Autumn 1	Students learn how to break problems into steps and represent solutions using flowcharts and pseudocode. They study common search and sort algorithms and compare how efficient they are.
10	Autumn 2	Students learn the core building blocks of programming: variables, data types, conditions and loops. They practise writing clear, robust code and learn how to test programs methodically.
10	Spring 1	Students learn how to build larger programs using functions, lists and external files. They practise defensive programming to make software safe and reliable. This unit strengthens their ability to plan, code and test.
10	Spring 2	Students apply subroutines, lists and file handling to create more robust and efficient programs. They learn defensive design techniques to make programs safer and more reliable.
10	Summer 1	In Summer 1, students focus on core GCSE Paper 1 content, studying systems architecture and memory in detail. They explore the components of the CPU, including registers and cache, and compare different types of primary and secondary storage.
10	Summer 2	In Summer 2, students study networks and cybersecurity as part of GCSE Paper 1. They examine different network topologies, key protocols and the layers involved in data transmission. Students also explore common cyber threats and how organisations protect data using encryption.
11	Autumn 1	Students learn how data is organised in real systems using tables, keys and relationships. They write SQL queries to search, sort and analyse data, and understand how validation keeps information accurate.
11	Autumn 2	Students explore real-world cyber threats and how organisations protect themselves. They learn about UK computing laws, ethical issues and the impact of technology on society.
11	Spring 1	Students refine their programming skills through targeted practice on algorithms, data handling and testing. They learn how to approach exam questions confidently and apply their knowledge under timed conditions.
11	Spring 2	Students refine their programming, tracing and problem-solving skills to meet GCSE exam requirements. They practise writing clear pseudocode, managing data structures and applying systematic testing.
11	Summer 1	In Summer 1, students undertake structured revision of GCSE Computer Science Paper 2 content, focusing on algorithms, programming techniques and data structures. They practise tracing, debugging and writing code.
11	Summer 2	Following the completion of GCSE examinations, students reflect on their progress and consolidate key computational thinking skills developed.