Farmor’s School Computing Department

“Those who can imagine anything, can create the impossible.” Alan Turing

Subject Head: Helen Bristow **Contact**: hbristow@farmors.gloucs.sch.uk

In a rapidly developing world, it has never been more important for our students to be technologically aware. We aim to equip our students to use computational thinking and creativity to understand and change the world. Computing has deep links with mathematics, science and design and technology, and provides insights into both natural and artificial systems.

During their time at Farmor’s, students will study the design, development and analysis of software and hardware used to solve problems in a variety of business, scientific and social contexts. They will learn how to stay safe online, develop their coding skills and improve their computational thinking, critical thinking, analysis, and problem-solving skills.

At KS3, students will follow a scheme of work based on the Teach Computing scheme. This has been built around an innovative progression framework where computing content has been organised into interconnected networks. The resources have been created by subject experts, using the latest pedagogical research and teacher feedback. KS4 & KS5 follow the AQA exam board specifications.

Farmor’s School Computing Department

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8
Year 7	Digital Literacy: Welcome to Farmor’s, intro to Satchel One, sending an email to a teacher. File management. Brief introduction to Microsoft applications: Word, Excel, Powerpoint.	Using media: Gaining support for a cause. Creating a digital product for a real-world cause.	Modelling data: Spreadsheets. Sorting and filtering data and using formulas and functions in spreadsheet software.	Programming essentials in Scratch part one: Applying the programming constructs of sequence, selection, and iteration in Scratch. Intro to robots.	Impact of technology: Collaborating online respectfully. Identifying how to use online collaboration tools respectfully. An introduction to the computing lab.	Networks: F rom semaphores to the internet. Recognising networking hardware and explaining how networking components are used for communication.	Programming essentials in Scratch: part two: Using subroutines to decompose a problem that incorporates lists in Scratch. Programming robots.	Vector Graphics: Design anything from logos and icons to posters, board games, and complex illustrations.
Year 8	Layers of Computing Systems: Simple Boolean logic and some of its uses in circuits and programming. The hardware and software components that make up computer systems. How instructions are stored and executed within a computer system.	Developing for the web: Searching. Threats. HTML and CSS.	Introduction to Python: Use Python programming to solve a variety of computational problems.	Representations: List examples and recall representations uses. Binary digits (bits) and natural numbers.	Mobile app development: Event handling. Sequencing. Variables. Selection. Operators.	Control systems with Flowol: Flowcharts, sequencing, sensors, subroutines, actuators and variables.	Digital Skills: File management, Bronze, Silver, Gold award – Microsoft applications, Staying safe online, Careers	
Year 9	Data Science: Using data to investigate problems and make real-world changes.	Python programming: Manipulating strings and lists. Creating a programming project.	Logic gates and binary revision/retrieval.	Introduction of cybersecurity: Identifying how users and organisations can protect themselves from cyberattacks.	SQL using Scratch-like interface: Creating accurate SQL statements and developing an understanding of SQL and database theory from normalisation to entity relationships.	Representations: Representing images and sound using binary digits.	Applying programming skills: Sensing and controlling with the micro:bit.	Digital Skills: File management, Bronze, Silver, Gold award – Microsoft applications, Staying safe online, Careers

Farmor’s School Computing Department

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9
Year 10	Programming - sequence: Determine the need for translators. Use sequence, variables, and input in Python. Design programs using a flowchart.	Computer Systems: Describe the role of the CPU. Explain the processes of the fetch-decode-execute cycle. Determine the role of main memory and secondary storage. Construct truth tables for three input logic circuits. Write a program using assembly language (LMC).	Programming - Selection: Use randomisation in programs. Work with arithmetic and logical expressions. Use selection and nested selection in Python.	Programming - Iteration: Use a while loop and a for loop in Python. Perform validation checks on data entry. Design programs using pseudocode.	Data representations: Explain how numbers, text, images, and sound are represented using binary digits. Perform operations on binary digits. Convert between units of measurement	Programming - Subroutines: Explain the differences between a procedure and a function. Describe scope of variables. Use functions and procedures as part of the structured approach to programming. Test a program for robustness.	Cyber security: Describe the various ways that users and organisations can be affected by cyberattacks. Demonstrate how organisations can prevent cyberattacks.	Algorithms 1 & 2: Define the terms ‘decomposition’, ‘abstraction’, and ‘algorithmic thinking’. Use trace tables. Describe a linear and binary search. Explain the key algorithms for a bubble, merge, and insertion sort.	Programming - Strings & Lists: Define the term ‘graphical user interface’ (GUI). Perform string handling operations. Describe the differences between a list and an array. Manipulate a list. Work with 2D lists.
Year 11	Networks: Explore how a computer network works from the hardware required to the protocols used for communication. Explore simulations of networks using Packet Tracer software.	Programming - Dictionaries & Data: Describe data structure, dictionaries, text files, CSVs and alternative approaches to programming solutions.	Relational databases and SQL: Describe a database and list its key terms. Determine the difference between a flat file and a relational database. Use structured query language (SQL) to retrieve and update data in a database.	Privacy and wider security: Determine the ethical, legal, environmental, and cultural impacts of technology.	Revision, assessment, feedback, programming practice.				

GCSE exam board: AQA 8525

Farmor's School Computing Department

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7
Year 12	**Fundamentals of programming:** Programming basics; selection, iteration, arrays, subroutines. Files and exception handling.	**Problem solving and theory of computation:** Solving logic problems. Structured programming. Writing and interpreting algorithms. Testing and evaluation. Abstraction and automation. Finite state machines.	**Data representation:** Number systems. Bits, bytes and binary. Binary arithmetic and the representation of fractions. Bitmapped graphics. Digital representation of sound. Data compression and encryption algorithms.	The Internet: Structure of the Internet. Packet switching and routers. Internet security. TCP/IP, standard application layer protocols. IP addresses. Client server model.	**Data structures:** Queues. Lists. Stacks. Hash tables and dictionaries. Graphs. Trees. Vectors.	**Communication:** Communication methods. Network topology. Client-server and peer-to-peer. Wireless networking, CSMA and SSID. Communication and privacy. The challenges of the digital age.	**Databases and software development:** Entity relationship modelling. Relational databases and normalisation. Introduction to SQL. Defining and updating tables using SQL. Systematic approach to problem solving.
Year 13	**Algorithms:** Recursive algorithms. Big-O notation. Searching and sorting. Graph-transversal algorithms. Optimisation algorithms. Limits of computation.	**Regular languages:** Mealy machines. Sets. Regular expressions. The Turing machine. Backus-Naur Form. Reverse Polish notation.	**OOP and functional programming:** Basic concepts of object-oriented programming. Object-oriented design principles. Functional programming. Function application. Lists in functional programming. Big data.	**Computer organisation and architecture:** Internal computer hardware. The processor. The processor instruction set. Assembly language. Input-output devices. Secondary storage devices.	**Hardware and software:** Hardware and software. Role of an operating system. Programming language classification. Programming language translators. Logic gates. Boolean algebra.		

A-level exam board: AQA 7517