Year 7

Unit Title	Unit Overview	Prior Knowledge / skills	New Learning
Unit 1 – Digital Literacy	You will explore the basic foundations of digital literacy and get to grips with their new school network.	• Basic hardware understanding • Basic hardware uses • Understanding of secure password and username needs	• Make use of the school network for day to day computational tasks • Awareness of basic formatting and word processing. • Make use of web browser facilities and an awareness of online safety
Unit 2 – Under the Hood	You will begin to learn about what exactly is inside of a computer and how the different parts work together to process data	• To be able to name basic input and output devices. • An understanding that the computer is a machine used to process data • An understanding of input, process and output.	• To name the components within the computer • To be able to explain how each of these devices interact • What affects the performance of the computer?
Unit 3 – Algorithms	You will drive into the world of computational thinking. An algorithm is a plan, a set of step-by-step instructions to solve a problem. If you can tie shoelaces, make a cup of tea, get dressed or prepare a meal then you already know how to follow an algorithm.	• To follow a basic instruction set • To be able to identify decisions in your everyday life • To understand the CPU and RAM and each use in a computer	• To be able to write a set of precise instructions to complete the task • To translate this into a flowchart • To understand the importance of pattern recognition and sub programs.
Unit 4 – Visual Programming	Discover the power of visual programming languages. Instead of writing code, you'll use visual	• Familiarity with simple problem-solving steps, such as breaking tasks into smaller parts.	• Learn to use block-based visual programming tools like Scratch to create functional programs.

	elements and blocks to create programs. Learn to design intuitive user interfaces, create animations, and develop interactive applications by connecting blocks together. Unlock the potential of coding through a graphical approach.	• Awareness of basic algorithms and how instructions can be sequenced to achieve specific outcomes.	• Introduction to designing simple, user-friendly interfaces that enhance the usability of applications. • Use visual programming to develop animations and interactive projects, connecting logic blocks to define behaviours.
Unit 5 – P5.JS Programming	Get hands-on with creative coding using P5.JS, a JavaScript library. You'll explore visual and interactive programming, creating animations, games, and interactive web experiences. P5.JS offers a simple syntax and powerful features, allowing you to bring your artistic and computational ideas to life in the browser.	• Fundamental coding concepts like variables, loops, and conditionals. • Prior experience with structuring instructions to solve problems or create simple programs	• Learn how to use p5.js, a JavaScript library, to create graphics, animations, and interactive programs in the browser. • Use simple functions to draw shapes, control colours, and create visual designs programmatically. • Implement user input (like mouse or keyboard events) to create dynamic and interactive web experiences.

Unit 6 is a short course unit in which students study e-safety requirements before the summer holidays. E-Safety content is interleaved into all units and homework.

Computer Science

The Ripley Academy

Year 8

Unit Title	Unit Overview	Prior Knowledge / skills	New Learning
Unit 7 – Intermediate Visual Programming	Take your visual programming skills to the next level. Build upon the basics and delve into more advanced concepts and techniques. Explore topics like event-driven programming, object-oriented design, and advanced user interface development. Create visually appealing and interactive applications using tools like Scratch.	• Familiarity with block-based programming tools like Scratch, understanding how to create simple programs by connecting blocks. • Experience with organizing blocks into sequences that follow logical steps to complete tasks.	• Learn how to build applications that respond to various user inputs and events dynamically, like mouse clicks, timers, and keyboard interactions. • Develop more sophisticated and visually engaging interfaces with custom layouts, animations, and interactive elements.
Unit 8 – Python Programming	Dive into the versatile world of Python programming. Learn the fundamentals of Python syntax, data types, control structures, and functions. Explore object-oriented programming concepts and modular design principles. Develop skills in file handling, exception handling, and working with libraries and modules. Gain practical experience in solving problems and building applications using the Python programming language.	• Familiarity with core concepts like variables, loops, and conditionals from prior exposure to programming (e.g., visual programming). • Understanding of how to approach problems by breaking them down into logical steps and writing basic algorithms.	• Master Python's basic syntax, including variables, loops, conditionals, and functions to write efficient code. • Gain experience using Python to implement fundamental coding concepts like variables, loops, and conditionals.

Unit 9 – Computer Logic	Discover the foundations of computer logic and digital circuits. Explore Boolean algebra, logic gates, and truth tables. Learn how to design and analyse combinational and sequential circuits. Dive into topics such as binary arithmetic, memory units, and computer organization. Gain a deeper understanding of how computers process and manipulate information at the fundamental level of logic.	• Familiarity with how computers follow a series of steps (algorithms) to solve problems. • Experience in breaking down complex problems and applying logical steps to reach solutions, such as in coding or mathematics.	• Understanding of the binary number system and how it is used to represent data in computers. • Learn the fundamentals of Boolean algebra, including AND, OR, and NOT operations, and how logic gates use these to perform computations. • Understand how computers perform binary addition and subtraction, and gain insight into memory units and how computers organize and store data.
Unit 10 – Graphics and Ethics	Examine the ethical considerations surrounding graphics and visual media. Explore the impact of digital manipulation, image rights, and copyright infringement. Discuss the ethical implications of using graphics in advertising, media, and entertainment. Analyse the role of graphic designers in promoting inclusive and ethical practices. Gain a deeper understanding of the ethical responsibilities in the creation and use of visual content.	• Exposure to how visual media influences perception and the role it plays in communication, advertising, and entertainment. • Familiarity with using simple graphic editing tools Paint to perform basic image manipulations (e.g., cropping, resizing, and colour adjustments).	• Explore how graphic designers can promote inclusive, honest, and ethical practices, ensuring that their work respects diverse audiences and maintains integrity in media and advertising. • Learn complex software for graphic editing tools like Adobe Photoshop or similar software to perform basic image manipulations (e.g., cropping, resizing, and colour adjustments).

Year 9

Unit Title	Unit Overview	Prior Knowledge / skills	New Learning
Unit 11 – Python Intermediate	Take your Python programming skills to the next level. Explore more advanced topics such as data structures, loops and algorithms. Develop proficiency in handling exceptions, working with files, and interacting with databases. Enhance your problem-solving abilities through hands-on projects and coding challenges.	• Familiarity with Python's basic syntax, including variables, loops, conditionals, and functions. • Experience in writing simple programs to solve basic problems using Python.	• Develop a deeper understanding of algorithm design, focusing on optimizing loops to solve more complex problems. • Gain experience using Python libraries, importing modules to extend Python's functionality for different tasks.
Unit 12 – HTML and the Web	Delve into the world of web development with HTML. Learn the fundamentals of HTML markup, including tags, attributes, and elements. Explore the structure and organization of web pages, creating layouts, adding multimedia content, and styling with CSS. Discover responsive design principles, accessibility considerations, and best practices for creating user-friendly websites. Gain hands-on experience in building and publishing web pages.	• Understanding of how to use a computer, browse the web, and navigate file systems. • Familiarity with how websites work, including the basic structure of a webpage and how browsers display content. • Experience in using text editors to create and modify content, such as basic documents or code snippets.	• Learn how to use HTML tags, attributes, and elements to structure web pages and create content such as headings, paragraphs, lists, and links. • Introduction to CSS for styling web pages, including layout, colors, fonts, and positioning to enhance the look and feel of a website.

Unit 13 – P5.JS Intermediate	Build upon your knowledge of P5.JS and further explore the creative possibilities of this JavaScript library. Dive deeper into advanced topics such as interactivity, animation, and responsive design. Learn to work with external data sources, create interactive visualizations, and integrate multimedia elements. Explore more complex coding concepts and techniques to enhance your ability to create engaging and interactive web experiences using P5.JS.	• Understanding of how to create simple shapes, animations, and basic interactive elements using P5.JS. • Familiarity with core JavaScript concepts like variables, loops, conditionals, and basic functions within the P5.JS environment. • Experience in building basic interactive projects that respond to user input, such as mouse and keyboard events.	• Explore more complex techniques for creating responsive, real-time interactivity and sophisticated animations using P5.JS. • Gain proficiency in incorporating multimedia elements like audio, video, and images into P5.JS projects, enhancing the user experience in interactive web applications.
Unit 14 – Cyber Security, morals and ethics	Explore the field of cybersecurity and the measures to protect computer systems and networks from unauthorized access, attacks, and data breaches. Learn about different types of cyber threats, such as malware, phishing, and social engineering.	• Familiarity with how computers and networks function, including the flow of data and basic network communication. • General understanding of safe online practices, such as creating strong passwords and avoiding suspicious links or emails. • Basic awareness of ethical issues in technology, including privacy and data protection.	• Learn about various cyber threats, including malware, phishing, and social engineering, and how they target vulnerabilities in computer systems. • Explore security techniques such as firewalls, encryption, and authentication methods to protect data and systems from unauthorised access and attacks. • Understand the principles of encryption, including symmetric and asymmetric encryption, and how they are used to secure data transmission and storage.
Unit 15 - Animation	Step into the captivating realm of animation. Learn the principles of animation, including timing, spacing, and movement. Explore different	• Familiarity with fundamental drawing techniques and design principles that support animation creation.	• Develop storytelling skills specifically for animation, learning how to effectively convey messages and emotions through

	animation techniques, such as traditional hand-drawn animation, stop motion, and computer-generated animation. Develop storytelling skills and create engaging animations that entertain and communicate effectively.	• Experience in basic narrative structures and how stories are conveyed through visual media. • Basic knowledge of digital art and design tools used for creating static images or graphics. Learn key animation principles such as timing, spacing, squash and stretch, and anticipation, to create fluid and realistic movements.	visual sequences and character movements. • Learn key animation principles such as timing, spacing, squash and stretch, and anticipation, to create fluid and realistic movements. • Explore various animation methods, including traditional hand-drawn animation, stop motion, and computer-generated animation, understanding their unique processes and applications.

GCSE Computer Science

Year 10

Unit Title	Unit Overview	Prior Knowledge / skills	New Learning
Unit 1 - Inside the Computer	Explore the inner workings of a computer system. Learn about the components that make up a computer, including the CPU, memory, storage, and peripherals. Understand how data is represented and processed within the computer. Dive into topics such as binary representation, Boolean logic, and the fetch-decode-execute cycle. Gain insight into the architecture and operation of computers at a fundamental level.	• Familiarity with using computers and understanding their role in everyday tasks.	• Learn about the primary components of a computer system, including the CPU, RAM, storage devices, and peripheral devices, and their roles in overall functionality. • Explore how data is represented within a computer, focusing on binary representation and the significance of bits and bytes in data storage and processing. • Dive into the fetch-decode-execute cycle to learn how the CPU processes instructions, detailing each step's role in executing programs and tasks. • Gain insights into how clock speed and cache memory influence CPU performance, and understand the impact of these factors on overall system efficiency.
Unit 2 - The workings of the CPU	Dive into the intricate workings of the Central Processing Unit (CPU). Learn about the components and functions of the CPU, including the control unit, arithmetic logic unit (ALU), and	• Familiarity with the general structure of a computer, including understanding the roles of the CPU, memory, and storage	• Learn about the key components of the CPU, including the control unit, arithmetic logic unit (ALU), and registers, and understand their specific roles in processing data.

	registers. Explore the fetch-decode-execute cycle and understand how instructions are executed. Gain insights into concepts such as clock speed, cache memory, and pipelining. Develop an understanding of how the CPU interacts with other components of a computer system.	• Experience with simple algorithms and how instructions are processed in programming.	• Explore the fetch-decode-execute cycle in detail, understanding how the CPU retrieves, interprets, and executes instructions step by step.
Unit 3 - Binary Logic	Explore the fundamentals of binary logic and its applications in computing. Learn about Boolean algebra and logic gates, including AND, OR, and NOT gates. Understand how these gates are combined to create more complex circuits. Explore truth tables and logical operations. Dive into topics such as logic gate diagrams, Boolean expressions, and simplification techniques. Develop skills in designing and analysing digital circuits using binary logic.	• Familiarity with the binary number system and how it differs from the decimal system. • Awareness of basic electronic components and how they function in simple circuits. • Basic knowledge of how programming logic can apply to problem-solving.	• Explore the various types of logic gates (AND, OR, NOT) and understand their functions and how they process binary inputs to produce outputs. • Understand how logic gates can be combined to create more complex digital circuits and how these circuits operate. • Develop skills in creating truth tables to represent the outcomes of logical operations and analyse how different inputs affect the outputs of circuits. • Learn to draw logic gate diagrams and apply Boolean simplification techniques to design and optimize digital circuits effectively.
Unit 4 - Data Representation	Gain an understanding of how data is represented and stored in computer systems. Explore different number systems, including binary, decimal, and hexadecimal. Learn about data formats such as ASCII and Unicode for representing characters. Dive into	• Familiarity with what data is and its importance in computing, including simple concepts of how data can be stored and accessed. • Awareness of how the binary number system functions and its role in computer operations.	• Explore different number systems, including binary, decimal, and hexadecimal, and understand how they are used in computing. • Dive into binary arithmetic, including addition, subtraction, and understanding

	topics like binary arithmetic, two's complement representation, and fixed-point and floating-point numbers. Understand the concept of data compression and its impact on storage efficiency. Develop skills in converting and manipulating data representations in computer systems		two's complement representation for negative numbers. • Learn about the concept of data compression, its various techniques, and its impact on storage efficiency, including how and why data is compressed for different applications.
Unit 5 - Algorithms	Dive into the world of algorithms and computational thinking. Learn the fundamental concepts and techniques used to solve problems algorithmically. Understand algorithm design strategies such as iteration, recursion, and divide-and-conquer. Explore algorithm efficiency and analysis, including Big O notation. Learn about searching and sorting algorithms, data structures, and algorithmic problem-solving techniques. Develop skills in designing, implementing, and analysing algorithms to solve real-world problems.	• Familiarity with fundamental programming concepts, such as variables, loops, and conditionals, which are foundational for understanding algorithms. • Experience in breaking down problems into smaller parts and developing logical approaches to find solutions.	• Learn the basic principles of algorithms, including their purpose in problem-solving and how they can be represented in pseudocode or flowcharts. • Explore various algorithm design strategies, including iteration, recursion, and the divide-and-conquer approach, and understand when to use each method effectively. • Learn about common searching (e.g., linear search, binary search) and sorting algorithms (e.g., bubble sort, merge sort and insert sort), including their implementation and efficiency.
Unit 6 - Computational Thinking	Develop essential computational thinking skills. Understand the core principles and strategies used to tackle complex problems. Explore abstraction, decomposition, pattern recognition, and algorithmic thinking.	• Familiarity with general problem-solving techniques and the ability to break down simple problems into smaller steps. • Awareness of fundamental programming concepts, which	• Understand the fundamental principles of computational thinking, including abstraction, decomposition, pattern recognition, and algorithmic thinking. • Learn to identify essential features of a problem while ignoring irrelevant details,

	Learn to break down problems into smaller, manageable parts and develop step-by-step solutions. Apply computational thinking techniques to a variety of contexts, such as data analysis, simulations, and automation. Develop critical thinking and problem-solving abilities that are applicable across various disciplines	provide a foundation for understanding computational approaches.	and break complex problems into smaller, manageable components for easier analysis and solution development. • Explore how to recognize patterns and trends within problems to apply existing solutions or develop new approaches. • Apply computational thinking techniques to real-world scenarios.

Year 11

Unit Title	Unit Overview	Prior Knowledge / skills	New Learning
Unit 7 - Defensive Design	Explore the principles of defensive design in software development. Understand the importance of designing software systems that are robust, secure, and resistant to errors and vulnerabilities. Learn techniques for handling exceptions, input validation, and error handling. Dive into topics such as defensive programming, code review, and testing strategies. Develop skills in writing resilient and secure code that	• Familiarity with fundamental programming concepts, including variables, control structures, and basic syntax in at least one programming language. • Awareness of the software development lifecycle and the importance of building functional applications.	• Understand the core principles of defensive design and its significance in creating robust and secure software systems that minimize vulnerabilities. • Learn techniques for effectively handling exceptions and errors, ensuring that programs can recover gracefully from unexpected situations without crashing. • Develop skills in various testing strategies, including unit testing and integration testing, to ensure that software behaves as

	can withstand unexpected situations and potential threats.		expected under different conditions and is resilient to failures and threats.
Unit 8 - System Software	Gain insight into the role and functions of system software in computer systems. Explore operating systems, their components, and their interaction with hardware and applications. Learn about memory management, process scheduling, file systems, and device drivers. Dive into topics such as system utilities, software updates, and system security. Understand the importance of system software in providing an efficient and reliable computing environment.	• Familiarity with using computer systems and understanding the distinction between hardware and software. • Awareness of what an operating system is and its fundamental purpose in managing computer resources. • Knowledge of how files and folders are organized and managed within a computer system.	• Gain insight into the essential functions of system software, particularly operating systems, in managing hardware and facilitating communication between applications and hardware components. • Explore the key components of operating systems, including the user interface, and system libraries, and how they work together to provide a seamless user experience. • Learn about memory management techniques, including how the operating system allocates and manages RAM, virtual memory, and the importance of efficient memory usage.
Unit 9 - Networking	Explore the fundamentals of computer networking. Learn about network architectures, protocols, and technologies. Understand the layers of the TCP/IP models. Explore topics such as IP addressing, subnetting, routing, and switching. Dive into network security, including firewalls, VPNs, and intrusion detection systems. Gain hands-on experience in configuring and troubleshooting networks. Develop an understanding of network management and the importance of	• Familiarity with how computers work and their basic components, including knowledge of hardware and software. • Awareness of what the Internet is and basic concepts of how data is transmitted online.	• Explore the core concepts of computer networking, including the purposes and benefits of networks in facilitating communication and resource sharing. • Learn about different network architectures, such as client-server and peer-to-peer models, and their respective use cases and advantages. • Gain knowledge of IP addressing schemes, including IPv4 and IPv6, and learn about subnetting techniques to create efficient network segments.

	effective communication and collaboration in network environments.		• Learn about network security concepts, including firewalls, VPNs, and intrusion detection systems, and understand strategies for protecting networks from unauthorized access and threats. • Familiarity with the concept of sending and receiving data, including an understanding of bandwidth and latency.
Unit 10 – Ethical and Moral Computing	Delve into the ethical and moral considerations related to computing and technology. Explore the impact of technology on society, privacy, and individual rights. Discuss topics such as intellectual property, copyright infringement, and digital rights management. Understand ethical issues related to data collection, surveillance, and artificial intelligence. Explore professional codes of ethics and the responsibilities of individuals working in the computing field. Develop critical thinking skills to navigate ethical dilemmas and make informed decisions in the realm of computing.	• Familiarity with fundamental computing concepts, including how technology is used in everyday life. • Awareness of what privacy means in the context of technology and data use, along with basic knowledge of personal data protection.	• Understanding of what intellectual property and copyright are, including the implications of using digital content without permission. • Discuss the positive and negative impacts of technology on social structures, communication, and personal relationships. • Understand ethical issues surrounding data collection practices, surveillance technologies, and the potential for misuse of personal information. • Learn how to assess the implications of technology-related choices on individuals and society, fostering a culture of ethical responsibility within the computing profession.