The Radclyffe School
"Working Together for Excellence"

Long Term Plan – Computing and Business

Vision: The curriculum is designed to allow students to have access to transferable skills through both Computer Science and Information Technology and make informed choices on how they pursue the subjects as they continue their learning journey through the school. Computer Science should be engaging and practical, it should encourage creativity and develop problem solving skills through a variety of tasks. As active, aware and informed participants in the digital world. We also aim for the students to leave The Radclyffe School as digital citizens with the skill set to keep their knowledge up to date in the ever-changing digital technology. Students should be able to go online and stay safe and understand what they need to do if they do not feel safe while on social media or online gaming.

	HT1	HT2	HT3	HT4	HT5	HT6	Threshold concepts
Year 10	1.2 Memory and Storage	2.4 Boolean Logic 1.1 Systems Architecture	1.3 Computer networks, connections and protocols	1.4 Network Security	1.5 Systems software	1.6 Ethical, Legal, Cultural and environmental impacts of digital technology	Know important concepts and can define key terms identifying advantages and disadvantages. Know the design of a computer and how a computer works.
Year 11	2.1 Algorithms	2.2 Programming Fundamentals	2.3 Producing Robust programs	2.5 Programming languages and IDE's	Revision of topics		Know how to create a computer program from start to finish using a variety of tools and techniques. Can evaluate and test a computer program and find errors within a program fixing these errors to make the program workable.

Curriculum Intent

Year 7	Year 8	Year 9	Year 10	Year 11
By the end of Year 7, students have developed basic ICT competencies and can use computers effectively to achieve an outcome. They understand the fundamentals of Computer Science — for example: binary, algorithms, and the importance of sequencing in both computational thinking and programming. This year lays the foundations by introducing students to key concepts and habits required for successful programming, using block-based tools such as Scratch. Online safety is revisited and reinforced from KS2 to reflect the ever-changing nature of technology.	By the end of Year 8, students are able to build on the concepts learned in Year 7 with more advanced computer science knowledge. They explore the purpose of computer components and develop the ability to justify which device or system is most appropriate in a given context. Students begin using text-based programming to create purposeful outcomes, deepen their understanding of the risks associated with digital technology, and extend their prior knowledge of binary to include binary addition and an introduction to hexadecimal.	By the end of Year 9, students are able to study topics that bridge the gap between Key Stage 3 and Key Stage 4. They develop a more advanced understanding of how computers work through the Von Neumann architecture and produce independent coding outcomes using advanced Python programming techniques. Students explore searching and sorting algorithms and examine how operating systems use this information. They also investigate common threats to systems and networks, alongside appropriate prevention methods.	By the end of Year 10, students will have a strong foundational understanding of how computer systems work. They will be able to describe key theoretical concepts such as systems architecture, memory and storage, networks, systems software, and cybersecurity. Students will understand how digital technologies impact society, considering ethical, legal, environmental, and cultural issues. While the primary focus is theory (Paper 1), students will also begin to develop core programming skills using Python. Throughout the year, students are encouraged to think critically about the role of technology and apply knowledge through discussions, low-stakes quizzes, and retrieval practice. This builds the knowledge base and thinking skills needed for success in both written exams and programming tasks.	By the end of Year 11, students will be able to apply computational thinking to solve problems using the Python programming language. They will confidently use key programming techniques including selection, iteration, arrays, procedures, and file handling. Students will design, trace, and evaluate algorithms, including searching and sorting, and write robust, efficient code using subroutines and structured logic. They will also understand how translators and IDEs support the programming process. Throughout the year, students continue to develop problem-solving confidence through regular retrieval practice, applied programming tasks, and deliberate use of worked examples. This enables students to consolidate both theory and practical skills in preparation for final exams and future study.