Fairfield High School Curriculum Overview – Years 9

Subject	**Computer Science**	**Why do we study these units in Year 9?**
Taught in rotation with Art & DT	14 – 16 Lessons per rotation	Year 9 in Computer Science builds on the skills developed in Years 7 and 8, moving students from block-based and web-based coding to text-based programming with Python. Students begin by learning the fundamentals of Python, including variables, input, selection, and iteration, and then apply these concepts to create interactive stories, quizzes, and games using the Turtle library for animation. Throughout the year, students also focus on developing problem-solving and computational thinking skills by designing, testing, and debugging their own programs. By completing projects such as a Zork-style text adventure, scoring systems, and interactive quizzes, they gain practical experience in planning, coding, and evaluating digital solutions. The unit concludes with an introduction to cybersecurity, helping students understand safe and responsible use of technology. By the end of Year 9, students will have a solid foundation in programming and computational thinking, preparing them for more advanced computer science concepts in Key Stage 4.
Setting	Mixed ability teaching in mixed tutor groups	

Students are encouraged to be Responsible Global Citizens through /content on… safe and responsible use of technology, including cybersecurity awareness, digital ethics, and understanding the impact of technology on society.
We ensure all students experience high challenge by differentiated Python programming tasks, designing and debugging interactive projects, and extending code with advanced features for students ready to tackle more complex problem-solving.
Literacy work includes the introduction of programming vocabulary related to Python, including variables, loops, selection statements, functions, and data structures, alongside terminology for debugging and computational thinking.

Innovation and Creativity opportunities developing interactive stories, quizzes, games, and animations with Python's Turtle library, allowing students to personalise projects and explore creative coding solutions.
Employability opportunities and skills are enhanced through project planning, debugging, problem-solving, computational thinking, and collaborative coding tasks, all of which build practical skills valued in digital and technical careers.

Unit	Unit title	Knowledge and Understanding/content	Skills	Assessment
1	**Introduction to Computing and Baseline Assessment**	Assess current understanding of programming, digital literacy, and problem-solving concepts. Introduction to Python as the primary programming language.	• Demonstrate existing programming and computational thinking skills. • Navigate Python development environment.	**Baseline assessment of knowledge and skills.**
2	Introduction to Python / Turtle Methods	Understand the basics of Python programming and how to control the Turtle module for visual outputs.	• Write Python programs using Turtle commands to create shapes and patterns.	**Practical exercises creating Turtle drawings.**
3	**Variables and Input / Interactive Story**	Learn about variables, data types, and user input. Understand how to use variables to store and manipulate data.	• Create interactive programs that accept input and use variables to control program flow.	**Mini-project: Interactive story using variables and input.**
4	Selection and Iteration / Zork Game Project	Explore conditional statements (if/else) and loops (while, for). Understand how selection and iteration control program flow.	• Implement selection and iteration in Python programs. • Design and develop a text-based adventure game.	**Practical project: Zork-style game demonstrating selection and iteration.**
5	**True/False Quiz / Scoring System**	Learn how to implement scoring and simple logic in programs. Understand validation and control flow for quizzes.	• Create a Python quiz with questions, scoring, and feedback.	**Practical project: Python quiz with scoring system.**

6	**Testing and Debugging / Extensions / Quiz Submission**	Understand the importance of testing code, debugging errors, and extending programs with additional features.	• Test, debug, and improve Python programs. • Add extensions to existing projects.	**Project assessment based on functionality, correctness, and extensions.**
7	**Evaluation / Online Assessment**	Learn to evaluate programs against requirements and understand online assessment tools.	• Critically assess programs for correctness, efficiency, and user experience. • Submit work through online assessment platforms.	**Evaluation report and online assessment submission.**
8	**Introduction to Cybersecurity**	Understand the fundamentals of cybersecurity, including threats, safe practices, and ethical considerations.	• Identify cybersecurity risks and apply best practices to protect data.	**Knowledge-based quiz or short task on cybersecurity principles.**