Subject Overview Computing – Year 11

During the year students will build upon the understanding that they have gained from Year 10. Students will develop an understanding of algorithms and their use in program design. Students will develop their programming knowledge using the Python programming language and learn how to make their programs robust. Students will apply Boolean logic to different programs and investigate the different programming languages and IDE's.

	Half Term 1	Half Term 2	Half Term 3	Half Term 4	Half Term 5	Half Term 6
Knowledge & Skills	**Unit 2.1 Algorithms (4 Lessons)** Five basic algorithms for searching and sorting data. Use of Pseudocode and flowcharts to represent algorithms **Unit 2.2 Programming Fundamentals (4 Lessons)** Key constructs in programming. Use of variables. The common arithmetic operators. The common Boolean operators AND, OR, NOT The use of data types Understand how to programmatically implement the key constructs **Assessment 1** • Algorithms • Programming Fundamentals	**Unit 2.2 Programming Fundamentals (8 Lessons)** The use of basic string manipulation. The use of basic file handling operations. The use of records to store data The use of SQL to search for data. The use of arrays (or equivalent) when solving problems, including both one-dimensional (1D) and two-dimensional (2D) arrays **Assessment 2** • Systems Architecture • Memory and Storage • Computer networks, connections and protocols • Network Security • Systems Software • Ethical, legal, cultural and environmental impacts of digital technology • Algorithms • Programming Fundamentals • Producing Robust Programs	**Unit 2.3 Producing Robust Programs (6 Lessons)** Robust programming methods. Implement testing plans and using trace tables. Use of logic to make decisions. Interpret logic diagrams and scenarios Logic diagrams **Unit 2.4 Boolean Logic (4 Lessons)** Use of logic to make decisions. Interpret logic diagrams and scenarios Logic diagrams Understand Boolean logic Apply Boolean logic Interpret Boolean logic diagram **Assessment 3** • Producing Robust Programs • Boolean Logic • Programming languages and Integrated Development Environments	**Unit 2.5 Programming languages and Integrated Development Environments (6 Lessons)** Types of programming languages and the scenarios in which they are used. Interpreters, Compilers and IDE. Techniques used to develop programming solutions Understand the differences in programming languages Be able to determine when and where to use Identify the features of programming languages **Revision & Exam**	**Revision & Exam**	

See next page for "Beyond The Curriculum"

Beyond The Curriculum	**Places to Visit:** Robotics labs (universities/research centres): Software development companies **Websites to Explore** Computer Science Unplugged Scratch (MIT) Codecademy / Khan Academy - Computer Science courses Online visualisation tools for sorting/searching algorithms **Wider Reading / Periodicals:** "Algorithms to Live By: The Computer Science of Human Decisions" by Brian Christian and Tom Griffith "Computational Thinking" by Jeannette M. Wing (original paper or articles discussing it Articles on AI and machine learning **Societies:** The Royal Academy of Engineering **Enrichment Clubs / Competitions / Trips:** Coding Club – IT2	**Places to Visit:** Software development companies University computer science departments **Websites to Explore:** Repl.it / Online Python Tuto GitHub / GitLab Stack Overflow **Wider Reading / Periodicals:** "Clean Code" by Robert C. Martin "The Pragmatic Programmer" by Andrew Hunt and David Thomas Developer blogs **Societies:** Local Python user groups / CoderDojos **Enrichment Clubs / Competitions / Trips:** Coding Club – IT2	**Places to Visit:** Electronics labs University computer science lectures/open days **Websites to Explore:** Logic gate simulators (online) Comparative analysis of programming languages Online compilers/interpreters The History of Programming Languages (various resources) **Wider Reading / Periodicals:** "Gödel, Escher, Bach: An Eternal Golden Braid" by Douglas Hofstadter Articles on quantum computing Essays on the philosophy of programming languages: **Societies:** IEEE Computer Society: **Enrichment Clubs / Competitions / Trips:** Coding Club – IT2	**Places to Visit:** Software development companies: University computer science departments **Websites to Explore:** Repl.it / Online Python Tutor GitHub / GitLab Stack Overflo Documentation for chosen programming language (e.g., Python documentation) Linting tools/code formatters **Wider Reading / Periodicals:** "Clean Code" by Robert C. Martin "The Pragmatic Programmer" by Andrew Hunt and David Thomas. **Societies:** Local Python user groups / CoderDojos **Enrichment Clubs / Competitions / Trips:** Coding Club – IT2		