

<u>Timeline</u>	<u>Topic</u>	<u>Key concepts and knowledge</u>	<u>Skills development</u>	<u>Rationale</u>
Autumn 1 7 lessons	1.4 – Network Security 1.5 – Systems Software 1.6 – Ethical, legal, Cultural and environmental impacts of digital technology 1.4.1 – Threats to computer systems and networks 1.4.2 – Identifying and preventing vulnerabilities 1.5.1 – Operating Systems 1.5.2 – Utility software 1.6.1 – Ethical, legal, cultural and environmental impact	Students know: <ul style="list-style-type: none"> • A wide range of threats to computers systems. • a wide range of measures to avoid or combat these threats. • the purpose of an operating system. • the 5 major roles that an operating system provides. • the purpose of ‘Utility Software’ in a computer system. • the roles of a variety of utility software. • a wide range of computer related issues. • the implications of legislation on computer use. Students know how to: <ul style="list-style-type: none"> • describe a wide range of system threats and system security measures. • explain how many of these treats and security measures work. • discuss the relative dangers that these threats pose. • describe the 5 major roles that an operating system provides. • explain why the roles of operating systems are important. • explain the role of the kernel • describe the roles of a variety of utility software. • explain the contexts in which the software might be used. • explain a wide range of issues that are related to computing technology. • discuss the positive and negative effects that computing technology has on our privacy, culture, ethics and on the environmental. 	Subject-specific skills: Python programming skills and computational thinking skills Exam technique Employability skills: Self-management Oracy Communication Literacy Numeracy Creativity Problem solving Summarize Recall Career links: Cyber security manager, Forensic computer analyst Systems analyst British Values: Rule of law – exploring legislation governing the safe and legal use of computers,	Building on year 10 learning of computer networks and protocols this half term learners will move to exploring threats to networks and ways businesses take to prevent these threats, as well as the role of operating systems and utility software. Students also asked to reflect on the wider implications of technology and computers and the impact this has on society and the environment

<p>Autumn 2 7 lessons</p>	<p>2.3 – Producing robust programs 2.5 – Programming languages and IDEs</p> <p>2.3.1 – Defensive design 2.3.2 - Testing 2.5.1 – Languages 2.5.2 - IDEs</p>	<p>Students know:</p> <ul style="list-style-type: none"> the various strategies that programmers use to ensure that their programs are robust: Input Validation, Input Sanitisation, Authentication, Maintainability the various errors that may occur during the development of software. the purpose of testing. the various testing strategies including iterative, black-box, white-box, acceptance, alpha and beta testing that source code has to be translated into machine code in order for the CPU to be able to process the program. the difference between how an interpreter and a compiler translate source code. the importance of assembly language and the role of an assembler. <p>Students know how to:</p> <ul style="list-style-type: none"> explain the importance of a defensive design and to be able to describe the various defensive design strategies. explain how commenting, indentation and formatting can all add to a defensive design. explain how whitelists and blacklists can help in the defensive design process. describe various types of error and various testing strategies. discuss the suitability of various testing strategies for given scenarios to explain why source code cannot be processed by the CPU to explain the differences between compilers and interpreters. explain why assembly language is still used in certain circumstances and to be able to explain what an assembler does. 	<p>Subject-specific skills: Python programming skills and computational thinking skills Exam technique</p> <p>Employability skills: Self-management Oracy Communication Literacy Numeracy Creativity Problem solving Summarize Recall</p> <p>Career links: Computer Programmer, Forensic computer analyst, UX designer, penetration tester</p>	<p>In previous units, students have explored programming techniques and threats to networks. This half term learners bring this knowledge together to look at how programmers ensure producing robust programs. Students are also provided with an opportunity to apply this knowledge and develop programming skills to ensure robust programs can be created. Learners can then consider how they test programs to ensure they are effective and efficient in their design. The half term is consolidated by looking at computer languages and how the Python language performs the same role as other text based programming languages and what implications that has on how the computer understands the data input.</p>
-------------------------------	--	--	---	--

<p>Spring 1 6 lessons</p>	<p>2.1 - Algorithms 2.1.3 – Searching and sorting algorithms Searching & Sorting practical programming skills</p>	<p>Students know:</p> <ul style="list-style-type: none"> • how the linear search algorithm works. • how the binary search algorithm works • how the ‘Bubble Sort’ algorithm works • how the ‘Insertion Sort’ algorithm works. • how the ‘Merge Sort’ algorithm works. • how to show the steps to sort a set of data using bubble sort. • how to read/trace the algorithm <p>Students know how to:</p> <ul style="list-style-type: none"> • describe the steps taken to perform both a linear search and a binary search. • compare the algorithms’ relative efficiencies. • write both the linear search and binary search algorithms. • demonstrate the steps required to sort a set of data using the bubble/merge/insertion algorithm. • read/trace the algorithm. • write the logical steps of a bubble/merge/insertion sort algorithm 	<p>Subject-specific skills: Python programming skills and computational thinking skills Exam technique</p> <p>Employability skills: Self-management Oracy Communication Literacy Numeracy Creativity Problem solving Summarize Recall</p> <p>Career links: Data analyst Systems analyst</p>	<p>Building on learning in year 9, students re-visit the concepts of searching and sorting algorithms but will move learning forward by exploring how the search and sort algorithms can look both as flowchart algorithms and ERL/Python programs. Students will have opportunity to develop practical programming skills in their application of their understanding of the different search and sort algorithms.</p>
-------------------------------	---	---	--	---

<p>Spring 2 6 lessons</p>	<p>Practical Programming Skills & Theory Revision</p>	<p>Students know:</p> <ul style="list-style-type: none"> • how to apply practical programming skills and knowledge to solve real-world problems <p>Students know how to:</p> <ul style="list-style-type: none"> • plan algorithms using both flowchart and ERL to solve problems • write increasingly complex algorithms in Python to solve problems • test programs to identify and correct error and make programs more efficient 	<p>Subject-specific skills: Programming skills computational thinking skills Exam technique</p> <p>Employability skills: Self-management Oracy Communication Literacy Numeracy Creativity Problem solving Summarize Recall</p> <p>Career links: Software developer Applications programmer Systems programmer Multimedia programmer Systems analyst Computer sales support Database administrator IT technical support officer</p>	<p>This will provide learners with a final opportunity for planning, writing and testing both flowchart and ERL algorithms and practice reading pseudocode. Learners will also revise Theory content and apply using real-life problems, whilst developing exam techniques.</p>
-------------------------------	---	---	---	---

<p>Summer 1 5 lessons</p>	<p>Final Revision</p>	<p>Students know:</p> <ul style="list-style-type: none"> • core learning from Component 1 and Component 2 learning <p>Students know how to:</p> <ul style="list-style-type: none"> • Answer exam style questions including “discuss” questions • Plan, write and correct algorithms using both ERL and flowchart algorithms 	<p>Subject-specific skills: computational thinking skills Exam technique</p> <p>Employability skills: Self-management Oracy Communication Literacy Numeracy Creativity Problem solving Summarize Recall</p> <p>Career links: Network Manager</p>	<p>This final half term before the GCSE exam season start will be dedicated to revising and re-capping core knowledge before their real exams. Learners will sit mock examinations of past papers and spend time reviewing teacher feedback and personal performance to focus revision on specific areas of weakness.</p>
-------------------------------	-----------------------	--	---	---