Curriculum Map: Computer Science Year 10

	Autumn	Spring	Summer
Content Declarative knowledge 'I Know'	1.3.2 Wired and wireless networks, protocols, and layers 1.4.1 Threats to computer systems and networks 2.2.3 Additional programming techniques (subtopics: 2D lists, file handling) 2.1.2 Designing, creating and refining algorithms (subtopics: more ERL, Logic errors)	1.4.2 Identifying and preventing vulnerabilities 2.3.2 Testing	1.5.1 Operating systems 1.5.2 Utility software 1.6.1 Ethical, legal, cultural and environmental impact
Skills Procedural Knowledge 'I know how to'	☐ Modes of connection: o Wired • Ethernet o Wireless • Wi-Fi • Bluetooth ☐ Encryption ☐ IP addressing and MAC addressing ☐ Standards ☐ Common protocols including: o TCP/IP (Transmission Control Protocol/Internet Protocol) o HTTP (Hyper Text Transfer Protocol) o HTTPS (Hyper Text Transfer Protocol Secure) o FTP (File Transfer Protocol) o POP (Post Office Protocol) o IMAP (Internet Message Access Protocol) o SMTP (Simple Mail Transfer Protocol) ☐ The concept of layers ☐ Forms of attack: o Malware o Social engineering, e.g. phishing, people as the 'weak point' o Brute-force attacks o Denial of service attacks o Data interception and theft o The concept of SQL injection Use of Python programming language to do the following: ☐ The use of basic file handling operations: o Open o Read o Write o Close ☐ The use of records to store data ☐The use of arrays (or equivalent) when solving problems, including two-dimensional arrays Identify the inputs, processes, and outputs for a problem ☐ Create, interpret, correct, complete, and refine algorithms using: o Pseudocode o Flowcharts o Reference language/high-level programming language ☐ Identify logic errors	☐ Common prevention methods: o Penetration testing o Anti-malware software o Firewalls o User access levels o Passwords o Encryption o Physical security Use of Python programming language to do the following: ☐ Validation ☐ The purpose of testing ☐ Types of testing: o Iterative o Final/terminal ☐ Identify syntax and logic errors ☐ Selecting and using suitable test data: o Normal o Boundary o Invalid o Erroneous ☐ Refining algorithms	☐ The purpose and functionality of operating systems: o User interface o Memory management and multitasking o Peripheral management and drivers o User management o File management ☐ The purpose and functionality of utility software ☐ Utility system software: o Encryption software o Defragmentation o Data compression ☐ Impacts of digital technology on wider society including: o Ethical issues o Legal issues o Cultural issues o Environmental issues o Privacy issues ☐ Legislation relevant to Computer Science: o The Data Protection Act 2018 o Computer Misuse Act 1990 o Copyright Designs and Patents Act 1988 o Software licences (i.e. open source and proprietary)
Strategies	• How the choice of network topology can have an impact on the network traffic and avoiding data collision	• Be able to recognise the threats to system security	• Able to identify differences between types of Operating Systems, and suggest the right one based on requirements

Conditional Knowledge 'I know when to'	• How topologies can be combined to create a more efficient network • How the choice of client server architecture vs peer to peer architecture can impact security in a network • What is the need for IP addressing of resources on the Internet, and how this can be facilitated by the role of DNS services • Be able to recommend one or the other types of connection choices between Wired and various Wireless options based on a given scenario • Explain the need for devices to have standards and protocols for effective communication. • Be able to identify the need for 2D lists, and when to use one instead of 1D lists • Recognise differences between logic and syntax errors in code, and be able to debug the code	• Explain how the size of the key has a big impact on the strength of an encryption algorithm • Explain how humans are often the weakest point in security • Explain the differences between, and be able to suggest the best methods of preventing system threats • Explain the important role that Testing plays in creating robust programs • Explain the role of test data • Explain why it is important for test data to include a range of criteria and types	• Explain how computers allow us to run more applications at one time than it's RAM can hold. • Explain how computers allow us to run multiple applications at seemingly the same time • Explain why printers always print documents on a first come first served order • Explain why managing user profiles is important and how it is done. • Explain why it is hugely important today to understand the ethical and cultural issues related to computer science technologies, and identify pros and cons, and help to solve the problems arising out of this. • How providing insufficient/biased data to AI programs can make AI not useful at the very least, and dangerous at worst. • Identify situations where use of robots can help humans and situations where they can cause harm. • Identify why privacy of data is an important issue in the online world, suggesting ways to keep data private. • Identify situations where laws that govern computer usage have been broken, and what those laws are.
Key Questions	✓ Compare benefits and drawbacks of wired versus wireless connection ✓ Recommend one or more connections for a given scenario ✓ The principle of encryption to secure data across network connections ✓ IP addressing and the format of an IP address (IPv4 and IPv6) ✓ A MAC address is assigned to devices; its use within a network ✓ The principle of a standard to provide rules for areas of computing ✓ Standards allows hardware/software to interact across different manufacturers/producers ✓ The principle of a (communication) protocol as a set of rules for transferring data ✓ That different types of protocols are used for different purposes ✓ The basic principles of each protocol i.e. its purpose and key features ✓ How layers are used in protocols, and the benefits of using layers; for a teaching example, please refer to the 4-layer TCP/IP model ✓ Produce simple diagrams to show: ▪ The structure of a problem ▪ Subsections and their links to other subsections ✓ Complete, write or refine an algorithm using the techniques listed ✓ Identify syntax/logic errors in code and suggest fixes ✓ Threats posed to devices/systems ✓ Knowledge/principles of each form of attack including: ▪ How the attack is used ▪ The purpose of the attack	✓ Understanding of how to limit the threats posed in 1.4.1 ✓ Understanding of methods to remove vulnerabilities ✓ Knowledge/principles of each prevention method: ▪ What each prevention method may limit/prevent ▪ How it limits the attack ✓ The difference between testing modules of a program during development and testing the program at the end of production ✓ Syntax errors as errors which break the grammatical rules of the programming language and stop it from being run/translated ✓ Logic errors as errors which produce unexpected output ✓ Normal test data as data which should be accepted by a program without causing errors ✓ Boundary test data as data of the correct type which is on the very edge of being valid ✓ Invalid test data as data of the correct type but outside accepted validation limit ✓ Erroneous test data as data of the incorrect type which should be rejected by a computer system ✓ Ability to identify suitable test data for a given scenario	✓ What each function of an operating system does ✓ Features of a user interface ✓ Memory management, e.g. the transfer of data between memory, and how this allows for multitasking ✓ Understand that: ▪ Data is transferred between devices and the processor ▪ This process needs to be managed and what this entails (e.g. the use of buffers when transferring data to a printer) ✓ User management functions, e.g.: ▪ Allocation of an account ▪ Access rights ▪ Security, etc. ✓ File management, and the key features, e.g.: ▪ Naming ▪ Allocating to folders ▪ Moving files ▪ Saving, etc. ✓ Understand that computers often come with utility software, and how this performs housekeeping tasks ✓ Purpose of the identified utility software and why it is required ✓ Technology introduces ethical, legal, cultural, environmental and privacy issues ✓ Knowledge of a variety of examples of digital technology and how this impacts on society ✓ An ability to discuss the impact of technology based around the issues listed ✓ The purpose of each piece of legislation and the specific actions it allows or prohibits ✓ The need to license software and the purpose of a software licence ✓ Features of open source (providing access to the source code and the ability to change the software)

		✓ Ability to create/complete a test plan	✓ Features of proprietary (no access to the source code, purchased commonly as off-the-shelf) ✓ Recommend a type of licence for a given scenario including benefits and drawbacks
Assessment topics	September assessment (baseline)	Mini tests Jan-Feb assessment (mid-year)	Mini tests Jun-Jul assessment (end of year)
Cross curricular links/Character Education	Logical Thinking, Problem solving.	Logical Thinking, Problem solving.	Problem solving, Resilience. Relevance of study to software industry practise (e.g. algorithm design, testing)