Curriculum Map **Subject: Computer Science** **Year Group: 9, 10, 11**

	Autumn – Year 9	Spring – Year 9	Summer – Year 9
Content- WHAT will be learned? What previous learning can be linked? Why this order/**sequence**?	**1. Python Programming** Reinforcement of procedural programming concepts using text based programming. (Python) (Sequencing, Iteration, Selection, use of variables) 1.1. Text-based programming 1.2. Python Syntax 1.3. Sequencing 1.4. Selection 1.5. Iteration 1.6. Algorithmic thinking 1.7. Flowcharts 1.8. Variables 1.9. Data types 1.10. Casting 1.11. Input/Process/Output 1.12. Using an IDE 1.13. Test plan	**2. Digital Data** Students investigate what is **digital/binary data** and how different form of information (e.g. numbers, text, pictures, sounds) are stored on computers in a binary form. 2.1. Binary Data 2.2. Storage units (bit, Byte, KB, MB, GB, TB, PB) 2.3. Binary Conversions 2.4. Hexadecimal Conversions 2.5. Binary Additions 2.6. Binary Shifts 2.7. Binary Masks 2.8. Logic Gates Diagrams 2.9. Truth Tables 2.10. Bitmap Pictures 2.11. Sound Sampling 2.12. Lossy/Lossless compression	**3. Algorithmic Thinking** Students develop their algorithmic thinking skills to use and design algorithms to solve a wide range of problems. 3.1. Searching algorithms 3.2. Linear search 3.3. Binary search 3.4. Sorting algorithms 3.5. Insertion sort 3.6. Bubble sort 3.7. Merge sort 3.8. File handling 3.9. Abstraction 3.10. Decomposition 3.11. Algorithmic thinking
Skills- What will be developed?	Problem solving through Trial & Error / Troubleshooting, Abstraction & Decomposition Algorithmic Thinking using flowcharts Problem solving using maths concepts including arithmetic calculations (percentages, MOD/DIV, areas and volumes), x-y coordinates, Boolean logic.	Students will learn how to manipulate binary numbers and hexadecimal numbers and investigate how numbers, text, pictures and sound files can be stored as binary data.	Algorithmic Thinking using flowcharts, pseudocode and trace tables. Problem solving through Trial & Error / Troubleshooting, Abstraction & Decomposition
Key 'How'/'Why' Questions- What **powerful knowledge** will be gained? What areas/themes/concepts will be explored?	Reinforcement of procedural programming concepts using text based programming. (Python) (Sequencing, Iteration, Selection, use of variables)	This unit revisit the "history of computer" science, following the chronological development of computers and digital information. Students will learn about: • Binary numbers / conversions • Hexadecimal Conversions • Binary Additions • Logic Gates and Truth tables • ASCII Code / Unicode • Bitmap Pictures • Sampling Sound	Reinforcement of procedural programming concepts using text-based programming. (Python) (Sequencing, Iteration, Selection, use of variables) Use of subroutines. Investigating key algorithms used for searching data (Linear search, binary search) and sorting data (Sorting Algorithms) Applying linear search algorithm using data stored in a CSV file (File handling operations)
SEND- how will support be seen? Seating plans? Simplified questions?	• Step by step demonstrations • Step by Step Video Clips • Python Syntax Helpsheet • Self-marking online quizzes (low stake assessment)	• Learning Grid (incl. Teacher's Notes, Video Clips, Exam Questions with Mark Scheme or Model Answers, save progress as RAG against each learning objective of the course) • Self-marking online quizzes (low stake assessment)	• Learning Grid (incl. Teacher's Notes, Video Clips, Exam Questions with Mark Scheme or Model Answers, save progress as RAG against each learning objective of the course) • Self-marking online quizzes (low stake assessment)
Assessment- What? Why?	• Subject Knowledge Assessment (x2)	• Subject Knowledge Assessment (x2)	• Subject Knowledge Assessment (x2)
What **memory for learning** skills will be required- modelling? Concrete answers? Retrieval?	• Teacher demonstrations. • PRIMM's approach: Predict-Run-Investigate-Modify-Make • Trial and Error Approach	• Teacher demonstrations • Students practice • Online Interactive Activities	• Teacher demonstrations. • PRIMM's approach: Predict-Run-Investigate-Modify-Make • Trial and Error Approach

		• Interleaving of key concepts	
Literacy- reading, extended accurate writing and oracy opportunities	• The importance of accurate syntax when writing code • Programming Terminology (focusing on algorithm, Sequencing, Selection and Iteration)	• Computer Science Terminology (focusing on digital data concepts)	• The importance of accurate syntax when writing code • Programming Terminology (focusing on algorithm, Sequencing, Selection and Iteration)
Numeracy/computing skills	• Arithmetic calculations using the standard arithmetic operators. Percentage calculations, area/volume calculations, use of x,y coordinates and angles, unit conversions (e.g. Imperial/Metric). • Solving mathematical problems using an algorithm.	• Binary calculations and conversions • Hexadecimal Conversion • Binary additions and binary shifts	• Arithmetic calculations using the standard arithmetic operators. Percentage calculations, area/volume calculations, use of x,y coordinates and angles, unit conversions (e.g. Imperial/Metric). • Solving mathematical problems using an algorithm.
Character development	Students becoming more independent problem solvers developing their capacity to identify and fix syntax and logic errors more independently using a trial-and-error approach as well as proactively tracing an algorithm and predicting the output of an algorithm.	Students develop their logical mind, working on logical problems including Boolean logic, logic gates and Truth tables.	Students becoming more independent problem solvers developing their capacity to identify and fix syntax and logic errors more independently using a trial-and-error approach as well as proactively tracing an algorithm and predicting the output of an algorithm.
Equality/Diversity opportunities	Early introduction of programming skills to break the misconceptions that "coding is for boys" **E-Safety Curriculum** Ethical debate on the use of AI in today's society (incl. discussion on the Turing Test and chatGPT)	The role of women from the early days of computer science. (Early computer scientists, role of women at Bletchley Park during WWII, impacts of Grace Hopper work on High Level languages)	Impacts of Grace Hopper's work on the development of high-level languages and translators (and the discovery of the first computer "bug")
Homework/Independent learning	• Weekly Subject knowledge lesson + quiz set on knowitallninja.com • Online coding practice (Edabit.com, 101computing.net)	• Weekly Subject knowledge lesson + quiz set on knowitallninja.com	• Weekly Subject knowledge lesson + quiz set on knowitallninja.com
CIAG coverage/links	Promoting careers in software development, web/app design/development	Promoting careers in software development, web/app design/development, hardware engineering	Promoting careers in software development, web/app design/development

	Autumn – Year 10	Spring – Year 10	Summer – Year 10
Content- WHAT will be learned? What previous learning can be linked? Why this order/**sequence**?	**4. Programming Project** 4.1. Requirements Analysis 4.2. Top modular design 4.3. Data Dictionary 4.4. Flowcharts 4.5. Implementation 4.6. Procedural programming 4.7. Programming constructs 4.8. Iterative testing 4.9. Code sanitisation 4.10. Code maintainability 4.11. Final Testing 4.12. Evaluation Students complete the **controlled assessment programming project** of the course (Scenario provided by the exam board) following the System Life Cycle: • Analysis (Identifying user requirements, Decomposition) • Design (Algorithmic Thinking, flowcharts, Data Dictionary) • Implementation (Iterative Development of solution, iterative testing, troubleshooting) • Testing (Iterative and final testing) • Evaluation Reinforcement of **procedural programming concepts** using text based programming. (Python) (Sequencing, Iteration, Selection, use of variables) Use of subroutines. File handling operations	**5. Hardware Concepts** 5.1. CPU architecture 5.2. Von Newman Architecture & registers 5.3. FDE Cycle 5.4. Clock speed 5.5. Multi-core 5.6. Cache memory 5.7. Embedded systems 5.8. Primary memory (RAM/ROM) 5.9. Virtual memory 5.10. Secondary memory 5.11. Optical storage devices 5.12. Magnetic storage devices 5.13. SSD storage devices 5.14. Storage capacity calculations **6. Software Concepts** 6.1. Operating Systems 6.2. Utilities 6.3. Defragmentation 6.4. Encryption 6.5. Incremental/full backups 6.6. Drivers 6.7. Application software 6.8. Translators (Compiler/interpreter) Students develop their subject knowledge regarding the **hardware & Software components of a computer systems**. They can identify different hardware & Software components of a computer system, describe their purpose and key characteristics, compare a range of input, output and storage devices to identify the most suitable devices for a given scenario. Students investigate storage units and storage requirements vs. capacity calculations.	**7. Network Concepts** 7.1. LAN vs WAN 7.2. Network Topologies 7.3. Client/Server Networks 7.4. Network components (Switch, hub, firewall, router, wap) 7.5. Types of servers 7.6. Wireless transmission (WiFi, 3G/5G) 7.7. Wired transmission (Ethernet Cable) 7.8. TCP Stack 7.9. TCP/IP Protocols 7.10. Packet Switching 7.11. IP Addresses 7.12. MAC Addresses 7.13. Web hosting & DNS Servers 7.14. Network Security (Threats and solutions to secure a network) **8. Legal, Ethical, Environmental Impacts of Computer Science** 8.1. Data Protection Act *and GDPR* 8.2. Copyright Design & Patent Act 8.3. Freedom of Information Act 8.4. Computer Misuse Act 8.5. Creative Commons Licensing 8.6. Environmental Impacts 8.7. Contribution to global warming 8.8. Ethical Impacts 8.9. Information reliability, Fake News 8.10. Digital Divide 8.11. Working conditions 8.12. Monitoring 8.13. Freedom of speech Students develop their subject knowledge regarding **network technologies & concepts**. Students discuss the **legal, ethical environmental impacts** of computer science making reference to a range of news stories. They identify both positive and negative impacts.
Skills- What will be developed?	Problem solving through Trial & Error / Troubleshooting, Abstraction & Decomposition, algorithmic Thinking using flowcharts	Students identify the key characteristics of different hardware components that make a computer system. Students investigate storage units and storage requirements vs. capacity calculations	Students evaluate the impacts of computer science on our every-day lives and on how society and businesses operate.
Key 'How'/'Why' Questions- What **powerful knowledge** will be gained? What areas/themes/concepts will be explored?	Reinforcement of **procedural programming concepts** using text-based programming. (Python) (Sequencing, Iteration, Selection, use of variables) Use of subroutines. File handling operations	Students develop their subject knowledge regarding the hardware & Software components of a computer systems. They can identify different hardware & Software components of a computer system, describe their purpose and key characteristics, compare a range of input, output and storage devices to identify the most suitable devices for a given scenario.	Students develop their subject knowledge regarding network technologies & concepts. They first investigate the different hardware components needs to set up a LAN. (Switch, Hub, WAP, NIC cards, Firewall, Router) and then investigate Internet concepts (Packet switching, IP Protocols concepts) Students also investigate a range of network threats and for each threat identify possible solutions to minimise the threat. Students discuss the legal, ethical environmental impacts of computer science making reference to a range of news stories. They identify both positive and negative impacts.

			They learn about the different legal acts relevant to computer science: • Data Protection Act *and GDPR* • Copyright Design & Patent Act • Freedom of Information Act • Computer Misuse Act • Creative Commons Licensing
SEND- how will support be seen? Seating plans? Simplified questions?	• Learning Grid (incl. Teacher's Notes, Video Clips, Exam Questions with Mark Scheme or Model Answers, save progress as RAG against each learning objective of the course)	• Learning Grid (incl. Teacher's Notes, Video Clips, Exam Questions with Mark Scheme or Model Answers, save progress as RAG against each learning objective of the course)	• Learning Grid (incl. Teacher's Notes, Video Clips, Exam Questions with Mark Scheme or Model Answers, save progress as RAG against each learning objective of the course)
Assessment- What? Why?	• Assessment of students' programming project • Subject Knowledge Assessment	• Subject Knowledge Assessment (x2)	• Subject Knowledge Assessment (x2)
What **memory for learning** skills will be required- modelling? Concrete answers? Retrieval?	• Teacher demonstrations. • PRIMM's approach: Predict-Run-Investigate-Modify-Make • Trial and Error Approach	• Teacher demonstrations • Students practice • Online Interactive Activities • Interleaving of key concepts	• Teacher demonstrations • Students practice • Online Interactive Activities • Interleaving of key concepts
Literacy- reading, extended accurate writing and oracy opportunities	• Students produce a technical documentation of their project	• Computer Science Terminology (focusing on computer hardware concepts)	• Computer Science Terminology (focusing on computer networks concepts) • Essay questions on the impacts of computer science on society
Numeracy/computing skills	• Arithmetic calculations using the standard arithmetic operators. Percentage calculations, area/volume calculations, unit conversions (e.g. Imperial/Metric). • Solving mathematical problems using an algorithm.	• Unit conversions (eg Bytes, KB, MB, GB, TB) • Storage Capacity calculations	• Unit conversions, Bandwidth requirements
Character development	• Students becoming more independent problem solvers developing their capacity to identify and fix syntax and logic errors more independently using a trial-and-error approach as well as proactively tracing an algorithm and predicting the output of an algorithm.	• Developing a curious mind with a willingness to understand how computers work when referring to technologies we use on a daily basis and revisiting the history of Computer Science.	• Students develop ethical consideration when evaluating the impacts of computer science and new technologies such as AI on society
Equality/Diversity opportunities	• Challenging stereotypes and misconceptions linked to the different IT job roles	• Roles and impacts of key computer scientists including Alan Turing, Ada Lovelace, Grace Hopper, John Von Neumann etc.	• Reflect on ethical issues and the impact of the digital divide on society.
Homework/Independent learning	• Weekly Subject knowledge lesson + quiz set on knowitallninja.com	• Weekly Subject knowledge lesson + quiz set on knowitallninja.com	• Weekly Subject knowledge lesson + quiz set on knowitallninja.com
CIAG coverage/links	Discussing different IT jobs based on the System Life Cycle: Business Analysist, System Designer, Software Developer, System Tester, Project Manager, Technical Author	Focus on careers in STEM, Electronics, Hardware Engineering	Promoting careers in hardware engineering, network design and cyber security.

	Autumn – Year 11	Spring – Year 11	Summer – Year 11
Content- WHAT will be learned? What previous learning can be linked? Why this order/**sequence**?	**Unit 2 - Algorithmic Thinking and Programming** Reinforcement of **procedural programming concepts** using text based programming. (Python) (Sequencing, Iteration, Selection, use of variables) Use of subroutines. Investigating key algorithms used for searching data (Linear search, binary search) and sorting data (Sorting Algorithms) Investigate approaches used to create more robust programs, testing strategies and the use of IDEs.	**Unit 1 - Computer System** Students develop their subject knowledge regarding the **hardware & Software components of a computer systems**. They can identify different hardware & Software components of a computer system, describe their purpose and key characteristics, compare a range of input, output and storage devices to identify the most suitable devices for a given scenario. Students investigate storage units and storage requirements vs. capacity calculations. Students investigate what is **digital/binary data** and how different form of information (e.g. numbers, text, pictures, sounds) are stored on computers in a binary form. Students will learn about: • Binary numbers / conversions • Hexadecimal Conversions • Binary Additions • Logic Gates and Truth tables • ASCII Code / Unicode • Bitmap Pictures • Sampling Sound Students develop their subject knowledge regarding **network technologies & concepts**. They first investigate the different hardware components needs to set up a LAN. (Switch, Hub, WAP, NIC cards, Firewall, Router) and then investigate Internet concepts (Packet switching, IP Protocols concepts) Students also investigate a range of network threats and for each threat identify possible solutions to minimise the threat. Students discuss the **legal, ethical environmental impacts** of computer science making reference to a range of news stories. They identify both positive and negative impacts. They learn about the different **legal acts** relevant to computer science: • Data Protection Act *and GDPR* • Copyright Design & Patent Act • Freedom of Information Act • Computer Misuse Act • Creative Commons Licensing	**Final Revisions on all aspects of the course**
Skills- What will be developed?	Problem solving through Trial & Error / Troubleshooting, Abstraction & Decomposition. Algorithmic Thinking using flowcharts, pseudocode and trace tables.	Students identify the key characteristics of different hardware components that make a computer system. Students investigate storage units and storage requirements vs. capacity calculations	
Key 'How'/'Why' Questions- What **powerful knowledge** will be gained? What areas/themes/concepts will be explored?	Reinforcement of procedural programming concepts using text based programming. (Python) (Sequencing, Iteration, Selection, use	Students revisit and make connections between key computer science concepts including:	

	of variables) Use of subroutines. Students revisit key concepts including: • Abstraction • Decomposition • Algorithmic thinking • Flowcharts • Pseudocode • Searching algorithms (Linear/Binary Search) • Sorting algorithms (Insertion, Bubble and Merge Sort) • Programming Constructs (Sequencing/Selection/Iteration) • Database Concepts and SQL Testing Strategies • IDEs • High Level Vs Low Level Languages • Translators	• Hardware & Software • Components of the CPU • Von Neumann Architecture • FDE Cycle • Primary & Secondary Memory • Input, output & storage devices • Number Systems: Binary and Hexadecimal • Character Sets (ASCII/UNICODE) • Bitmap Pictures • Sound Sampling • Computer Network Concepts • Network Topologies • Network Hardware • TCP/IP Protocols • Network Security Concepts • Legal, Ethical, Environmental Impacts of Computer Science	
SEND- how will support be seen? Seating plans? Simplified questions?	• Learning Grid (incl. Teacher's Notes, Video Clips, Exam Questions with Mark Scheme or Model Answers, save progress as RAG against each learning objective of the course) • Video Clips to support home learning • Structured lessons and resources • Exam Revision Techniques	• Learning Grid (incl. Teacher's Notes, Video Clips, Exam Questions with Mark Scheme or Model Answers, save progress as RAG against each learning objective of the course) • Video Clips to support home learning • Structured lessons and resources • Exam Revision Techniques	
Assessment- What? Why?	• Subject Knowledge Assessment (x2)	• Subject Knowledge Assessment (x2)	
What **memory for learning** skills will be required- modelling? Concrete answers? Retrieval?	• Teacher demonstrations • Students practice • Online Interactive Activities • Interleaving of key concepts	• Theory concepts covered using lesson powerpoints and using written tasks in books • Regular low-stake assessment / retrieval practice using online and written quizzes • Key focus on terminology (Word based activities) • Techniques to create revision materials (mindmpas, flash cards) • Exam Techniques	
Literacy- reading, extended accurate writing and oracy opportunities	• Exam Paper Practise • Tips to answer Identify/Describe/Explain/Compare/Evaluate exam questions • Knowledge/Application/Evaluation writing frames • Producing exam revision materials extracting short/factual facts and definitions	• Exam Paper Practise • Tips to answer Identify/Describe/Explain/Compare/Evaluate exam questions • Knowledge/Application/Evaluation writing frames • Producing exam revision materials extracting short/factual facts and definitions	
Numeracy/computing skills	• Arithmetic calculations using the standard arithmetic operators. Percentage calculations, area/volume calculations, unit conversions (e.g. Imperial/Metric). • Solving mathematical problems using an algorithm.	• Unit conversions (eg Bytes, KB, MB, GB, TB) • Storage Capacity calculations • Binary and Hexadecimal conversions • Binary additions and shifts	
Character development	• Taking ownership of resources available, organising a revision timetable and creating revision materials to become a more resilient, resourceful and aspirational learner	• Taking ownership of resources available, organising a revision timetable and creating revision materials to become a more resilient, resourceful and aspirational learner	
Equality/Diversity opportunities	• Roles and impacts of key computer scientists including Alan Turing, Ada Lovelace, Grace Hopper, John Von Neumann etc.	• Understanding of the impacts of the DPA to prevent discrimination in the work place and in society. • Understanding the causes and impacts of the digital divide on our society. • Evaluating other ethical impacts of computer science on society	

Homework/Independent learning	• Weekly Subject knowledge lesson + quiz set on knowitallninja.com	• Weekly Subject knowledge lesson + quiz set on knowitallninja.com	
CIAG coverage/links	Promoting careers in software development, web/app design/development	Promoting careers in hardware engineering, network design and cyber security.	