Rutlish School

CURRICULUM MAP

Subject	**Computer Science**
Head of Department	**Ms S Salad**

SCHOOL INTENT

Rutlish School: Curriculum Intent

Rutlish School provides a meaningful, broad and balanced curriculum which is accessible to all, as well as supports and challenges all students. Through all we do, we prepare students for opportunities, responsibilities and experiences later in life for them to be aware of their responsibilities and feel confident to participate and contribute to society. We aim to inspire, enable and facilitate lifelong learners that build on their individual strengths and capabilities and achieve their ambitions. We seek to support our students in becoming healthy, happy, successful modern people young adults, who are knowledgeable, kind, aware, confident, capable and skilful members of society.

The school aims to:

- ensure that the curriculum is designed for every student of every ability and every background to be supported in making the best possible progress and attainment from their starting point; all students can access the curriculum offer, with planning and teaching that support, stretch and challenge all learners across a full range of abilities, and making any reasonable adjustments required where particular needs are identified;
- provide a curriculum that is sequenced to build skills and knowledge throughout students' time at Rutlish School, to promote a deeper understanding of the world outside the classroom and equip them for their next steps in education, careers and in life;
- ensure that our curriculum offer support different educational and career pathways, including EBACC and vocational;
- ensure our curriculum consistently promotes high moral standards, social and self-awareness and allows students to formulate informed opinions on social issues such as, equality, diversity and inclusivity as well as the practical aspects of society;
- enrich the curriculum and provide opportunities for students to build cultural capital, enhance a wide range of skills and knowledge beyond requirements of the national curriculum, and personalise and apply learning in other contexts;
- provide students with the skills and knowledge necessary to becoming independent, analytical, critical, and innovative thinkers and encourage students' curiosity, creativity, self-expression, resilience, and confidence; provide consistent opportunities for students to develop and enhance their reading skills, and support is provided to ensure all students are able to access he curriculum.

DEPARTMENT INTENT

The aim of Computer Science at Rutlish is to provide a high-quality computing education which equips students to use computational thinking and to creatively understand and change the world. We study Computer Science to help us think in a more logical way and become better at making decisions and solving problems in a world that is rapidly changing and demanding technological expertise. The curriculum will teach students key knowledge about how computers and computer systems work, and how they are designed and programmed.

KEY STAGE 3 RATIONALE/ INTENT

The KS3 curriculum has been designed to ensure learners have sufficient knowledge to stay safe online and use computers safely in life. The KS3 curriculum also provides a focus on developing resilient learners who are able to learn from mistakes and effectively solve problems. The topics at KS3 give a basis of knowledge, skills and understanding to allow students to progress onto either i-Media or Computer Science at KS4 and will provide exposure to those subjects so that students can make an informed decision on their GCSE choices.

KEY STAGE 4 RATIONALE/ INTENT

At KS 4 students build on the skills they have learned in KS 3. The curriculum develops the student's ability to become a more resilient learner that is focused on logical thinking, decomposition and abstraction of problems to ultimately become advanced at solving computational problems. Students also continue to develop skills in digital literacy where they use computer systems to create and evaluate digital products for specific audiences and purposes.

KEY STAGE 5 RATIONALE/ INTENT

The KS 5 program of study develops the ability of students to become independent learners and to use research to extend their knowledge beyond the scope of the specification. Computer Science at KS 5 provides students with the challenge of programming a computer game that embeds complexities, creativity and allows students to problem solve and follow a test driven development process. The curriculum enables students to become critical thinkers that are able to apply and evaluate their knowledge to different scenarios. The curriculum enables students to become critical thinkers that are able to apply and evaluate their knowledge to different scenarios.

QUALITY FIRST TEACHING

- Scaffolding, modelling and sample answers incorporated into all lesson resources.
- All programming tasks and theory questions set in GCSE exam question format
- Metacognition strategies incorporated into lessons through providing opportunities to peer and self-assess and reflect on progress.
- Explicit teaching of programming and computational skills to enable students to become independent learners focused on problem solving.
- Continuous recap and revise promote knowing more and remembering more.
- Self-assessment to promote independent learning, self-correction and developing individual subject knowledge; algorithmic quizzes targeted at student weaknesses.

Rutlish School

YEAR 10: Computer Science: THEORY						
	Autumn 1	**Autumn 2**	**Spring 1**	**Spring 2**	**Summer 1**	**Summer 2**
KNOWLEDGE	**UNIT OF WORK:** **Systems Architecture**	**UNIT OF WORK:** **Data Representation**	**UNIT OF WORK:** **Image and Sound Representation, Compression**	**UNIT OF WORK:** **Networks and Protocols**	**UNIT OF WORK:** **Network Security**	**UNIT OF WORK:** **Ethical Legal Cultural**
	The purpose of the CPU, the common components and their functions. The different factors that affect CPU performance and how primary and secondary memory is used in computers. The need for secondary storage and the advantages and disadvantages of different types of storage.	How data is represented in binary and hex. To recall knowledge from KS 3 to convert binary to denary, add and subtract binary numbers. What ASCII is and to be able to recall knowledge from KS 3 on ASCII representation. To perform logic and arithmetic shifts.	Explore how images are represented as specific code and what the effects of images size and quality is in relation to bit depth. How analogue sound are represented in binary. What lossy and lossless compression is?	Explore the different types of networks and topologies and the hardware needed to connect to different types of networks. To explore wired and wireless networks and to understand the different protocols that operate at each layer of the different types of networks.	Explore how networks come under attack and how networks can be protected from threats. To be able to identify the type of network threats and vulnerabilities. To explore the different types of OS and to understand the different purposes and functions of an OS. Types of utility software and the purpose and function of each.	Impacts of digital technology on wider society. Explore the legislation relevant to Computer Science.
KEY SKILLS	Analyse the performance of the CPU. Remember and understand the architecture the CPU. Remember and understand the purpose and characteristics of embedded systems. Evaluate the use of secondary storage to a given scenario.	Understand the methods of conversions. Apply the formulae to various problems.	Analyse graphs – identify how sound are stored in digital format. Understand the effect of sample rate on sound quality file size. Apply formulae to calculate file sizes. Understand and remember lossy and lossless compression and be able to apply the correct method of compression to a given scenario.	Remember and understand the different types of network protocols and be able to recall the different network layers and their functions. To be able to apply formulae to calculate network speed. To be able to evaluate the advantages and disadvantages of different networks and topologies. Create summaries of flipped learning materials – create your own questions, develop independent learning.	Remember and understand each of the types of threats to a network. Remember and understand the types of utility software. Understand how to identify and prevent network vulnerabilities.	Apply the relevant legislation to a given scenario. To understand and remember the impact of technology on society. To be able to evaluate the impact of technology on society, culture and ethics. Discuss the impacts of digital technology on the wider society including ethical issues, cultural issues and environmental issues. Describe the features of open source and proprietary software licences.
HOW DO WE BUILD ON SKILLS AND KNOWLEDGE?	**Link to KS 3:** Year 7 : Introduction to computers. Year 8 : How computers work.	**Link to KS 3:** Year 7 : Introduction to binary. Year 8 : Data representation (build on year 7 skills + compression and image representation) Year 9 - Build on year 8 recap knowledge of year 7 and 8 + intro to Two's complement and S & M, recap on ASCII.	**Link to KS 3:** Year 8 - Compression algorithms topic.	**Link to KS 3:** Year 8 - networks topic	**Link to KS 3:** Year 8 - networks topic KS 3 – Online safety topics	**Link to KS 3:** Year – Ethical and legal

Rutlish School

LINKS TO THE WORLD **i.e. links to careers; equality: gender, class, ethnicity, etc.; different subjects**	**Careers:** Link to Jon Von Neuman and Alan Turing – see PPT.	**Careers:** Alan Turing – equalities - see ppt. **Cross curricular** Link to Maths – numeracy.	**Careers:** See ppt slide on links to careers in web design and graphic design. **Cross curricular** Link to Maths – numeracy. Link to English - extended writing questions.	**Careers:** See ppt slide on links to careers in networking jobs **Cross curricular** Link to Maths – numeracy. Link to English - extended writing questions.	**Careers:** See ppt slide on links to careers in cyber intelligence officer. **Cross curricular** PSHE – online safety	**Cross curricular** PSHE & Geography
ASSESSMENTS **Summative and Formative as applicable**	**Formative:** Self and peer assessment Class discussions Classwork and homework Exit tickets Think pair share Quizzes Mnemonics **Summative:** End of unit test	**Formative:** Self and peer assessment Class discussions Classwork and homework Starter questions Exit tickets Think pair share Quizzes Mnemonics **Summative:** End of unit test	**Formative:** Self and peer assessment Class discussions Classwork and homework Exit tickets Think pair share Quizzes Mnemonics Retrieval power Ticket **Summative:** End of unit test	**Formative:** Self and peer assessment Class discussions Classwork and homework Exit tickets Think pair share Quizzes Mnemonics Retrieval power Ticket **Summative:** End of unit test	**Formative:** Self and peer assessment Class discussions Classwork and homework Starter questions Exit tickets Think pair share Quizzes Mnemonics Retrieval power Ticket **Summative:** End of unit test	**Formative:** Self and peer assessment Class discussions Classwork and homework Exit tickets Think pair share Quizzes Mnemonics Retrieval power Ticket **Summative:** End of unit test
FEEDBACK SUPPORTS LEARNING	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓
SPECIALIST VOCABULARY	CPU RAM ROM Optical Magnetic Solid state Primary and secondary storage Cache Cores Clock speed	Bit Byte Kilobyte Megabyte Gigabyte Overflow Two's complement Sign and Magnitude.	Lossy Lossless Bit depth Binary Sample rate Compression	Wireless Wired Protocol IP address Encryption HTTP, HTTPS, FTP,POP, MAP, SMTP, TCP Routers, switches WAP, The cloud Star, Mesh	Malware Phishing Brute force attack DOS attack SQL Injection Pen test Firewalls User access levels Passwords Encryption	Ethical Legal Cultural Environmental Privacy Data Protection Act Computer Misuse Act Copyright Designs and Patents Act Software licences.
QUALITY FIRST TEACHING	✓ Strategies to learn more, remember more (metacognition) used in lessons e.g. retrieval, elaboration, interleaving, dual coding, etc. ✓ Differentiation and reasonable adjustments for students with SEND, EAL, etc. such as scaffolding, visual aids, audio, physical resources, planned questioning, etc. ✓ Opportunities for Literacy, Numeracy and Oracy, including a focus on reading ✓ Opportunities to apply key concepts and address misconceptions					

Rutlish School

YEAR 10: Practical						
	Autumn 1	**Autumn 2**	**Spring 1**	**Spring 2**	**Summer 1**	**Summer 2**
KNOWLEDGE	**UNIT OF WORK:** **COMPUTATIONAL THINKING, ALGORITHMS AND PROGRAMMING**	**UNIT OF WORK: Algorithms & Pseudo Code**	**UNIT OF WORK: Subprograms**	**UNIT OF WORK: Subprograms**	**UNIT OF WORK: Lists and loops**	**UNIT OF WORK: Nested Lists & File Reading**
	To know how to use Python to code small programs from the given tasks and will be able to apply knowledge to new programming scenarios. Students will know how to use Python to code programs using the key skills listed below.	Students will be able to solve computational problems using flowcharts and pseudo code. They will be able to use trace tables to trace through a program to identify logic errors and to trace data at different stages of a program. They will understand Boolean logic and will be able to recall the rules for AND, NOT and OR gates	Through practical programming students will be able to use and recall their knowledge of selection and iteration and apply it to code programs using functions and procedures. Understand the difference between a function and a procedure and learn to organise their subprograms into larger programs that solve problems.	Through practical programming students will be able to use and recall their knowledge of selection and iteration and apply it to code programs using functions and procedures. Understand the difference between a function and a procedure and learn to organise their subprograms into larger programs that solve problems	Through practical programming students will be able to use and recall their knowledge of index positions, selection, and iteration and apply it to code programs that makes use of lists. Students will understand the difference between 1 dimensional and 2 dimensional arrays. They will develop solutions to larger programs using lists.	Through practical programming students will be able to use and recall their knowledge of lists to create programs using nested lists. Students will build their existing knowledge to code a program that can read and write data to a file. The will apply all their programming knowledge to effectively solve a complex problem.
KEY SKILLS	**Use Python to apply the following practical skills: (Recall embed skills from KS 3)** Escape sequences Code small programs Identify and correct syntax errors independently Maths operators Data type, variable and inputs String formatting Relational operators Selection	**Use Python to apply the following practical skills:** To be able to: Use the correct flowchart symbols to accurately sequence the steps to solve a problem. Use .draw.io software to create flowcharts. Use normal data, boundary data and erroneous data to test a program. Use a trace table effectively to trace the content of data in variables. Use the OCR Exam ref language to write and read pseudo code. Draw Boolean gates and tables.	**Use Python to apply the following practical skills:** Decomposition Breaking larger programs down into subprograms. Difference between functions and procedures. Calling function/procedure Return a value to a function	**Use Python to apply the following practical skills:** Decomposition Breaking larger programs down into subprograms. Difference between functions and procedures. Calling function/procedure Return a value to a function	**Use Python to apply the following practical skills:** For loops While loops Built in functions Validation in programs Sub programs	**Use Python to apply the following practical skills:** Reading and writing to a file Nested lists Flowcharts Pseudo code
HOW DO WE BUILD ON SKILLS AND KNOWLEDGE?	**KS 3 SOW** Year 7 – Input, Output, variables, selection, data types	**KS 3 SOW** Practical programming using OR, AND to code programs making use of Boolean logic e.g. grade calculator, password program.	**KS 3 SOW** Year 9 SOW – selection and iteration.	**KS 3 SOW** Year 9 SOW – selection and iteration	**KS 3 SOW** Year 9 SOW – sorting and searching algorithms	**KS 3 SOW** Iteration Selection Decomposition

Rutlish School

LINKS TO THE WORLD i.e. links to careers; equality: gender, class, ethnicity, etc.; different subjects	**Careers:** See PPT slides. Famous programmers from the past. Companies that use Python Programming software used in the world of work.	**Careers:** George Boole – see ppt slide. – English Computer Scientist.	**Careers:** Reference should be made to careers in programming. Careers in coding, software development and programming - icould	**Careers:** Reference should be made to careers in programming. Careers in coding, software development and programming - icould	**Careers:** Reference should be made to careers in programming. These can include: software development, network security, networking, AI.	**Careers:** Reference should be made to careers in programming. These can include: software development, network security, networking, AI.
ASSESSMENTS Summative and Formative as applicable	**Formative:** Self-evaluation of programs Class discussions Think pair share Peer feedback **Summative:** End of unit test	**Formative:** Self and peer assessment of trace tables, pseudo code and flowcharts. Think pair share **Summative:** End of unit test	**Formative:** Self-evaluation of programs Class discussions Think pair share Peer feedback **Summative:** End of unit test	**Formative:** Self-evaluation of programs Class discussions Think pair share Peer feedback **Summative:** End of unit test	**Formative:** Self-evaluation of programs Class discussions Think pair share Peer feedback **Summative:** End of unit test	**Formative:** Self-evaluation of programs Class discussions Think pair share Peer feedback **Summative:** End of unit test
FEEDBACK LESSONS PLANNED	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓
SPECIALIST VOCABULARY	Escape sequences Data types Variables String formatting Relational operators Input/Output IF ELSE	Algorithms Boolean Pseudo code Abstraction Trace table Flowchart Input, Process, Output Decision	Function Procedure Parameter Argument Return Call Decomposition Len Min Max Scope Global vs Local Types of errors – Syntax, Logic, Runtime	Function Procedure Parameter Argument Return Call Decomposition Len Min Max Scope Global vs Local Types of errors – Syntax, Logic, Runtime	For loop While loop Lists Array Functions Subprograms	Reading and writing to a file Decomposition Readline Writeline Try except else
QUALITY FIRST TEACHING	✓ Strategies to learn more, remember more (metacognition) used in lessons e.g. retrieval, elaboration, interleaving, dual coding, etc. ✓ Differentiation and reasonable adjustments for students with SEND, EAL, etc. such as scaffolding, visual aids, audio, physical resources, planned questioning, etc. ✓ Opportunities for Literacy, Numeracy and Oracy, including a focus on reading ✓ Opportunities to apply key concepts and address misconceptions					

Rutlish School

YEAR 11: Programming project & Theory						
	Autumn 1	**Autumn 2**	**Spring 1**	**Spring 2**	**Summer 1**	
KNOWLEDGE	**UNIT OF WORK: Algorithms-SECTION 6**	**UNIT OF WORK: Programming project**	**UNIT OF WORK: Programming project**	**UNIT OF WORK: Revision**	**UNIT OF WORK: Revision**	**UNIT OF WORK: End of course**
	Understand and apply computational thinking methods. Understand what linear and binary search algorithms are. Understand what bubble, insertion and merge sort algorithms are.	Introduce students to the programming project: Students will know and understand the outline of the program they have to code and will be able to brainstorm to decompose the problem.	Students will know and understand the outline of the program they have to code and will understand how to create a test table.	Recap subject knowledge for each topic using know more remember more strategies:	Recap subject knowledge for each topic using know more remember more strategies:	
KEY SKILLS	**Be able to:** Trace linear and binary search algorithms. Trace bubble, insertion and merge sort algorithms. Produce algorithms using flowcharts and pseudo code. Interpret correct or complete algorithms.	**Be able to:** Decompose the problem, write up the Analysis, plan the program using flowcharts and pseudo code. Code part 1 of the program.	**Be able to:** Code part 2 of the program. Create test tables. Test the program.	Study techniques enhanced Creating knowledge organisers Using exam reference language to write and read program code.	Study techniques enhanced Creating knowledge organisers Using exam reference language to write and read program code.	
HOW DO WE BUILD ON SKILLS AND KNOWLEDGE?	**Link to KS 3:** Year 9 SOW sorting algorithms	**Link to KS 3 & year 10:** Python programming skills learned in years 7 – 10.	**Link to KS 3:** Year 9 SOW sorting algorithms	**Revisit subject knowledge:** Section 1 Section 2 Section 3 Section 4 Section 5 Section 6 Programming revision theory	**Revisit subject knowledge:** Section 1 Section 2 Section 3 Section 4 Section 5 Section 6 Programming revision theory	
LINKS TO THE WORLD i.e. links to careers; equality: gender, class, ethnicity, etc.; different subjects	**World of work:** Explicit reference to how the searching and sorting algorithms are used in the world of work.	**World of work:** Explicit reference to how the searching and sorting algorithms are used in the world of work.	**World of work:** Explicit reference to how the searching and sorting algorithms are used in the world of work.			

Rutlish School

ASSESSMENTS Summative and Formative as applicable	**Formative:** Self and peer assessment Class discussions Classwork and homework Exit tickets Think pair share Quizzes Mnemonics **Summative:** End of unit test	Independent programming project.	Independent programming project.	**Retrieval practice** Interleaving Spaced practice Dual coding Quizzes **Summative** Revision tests	**Retrieval practice** Interleaving Spaced practice Dual coding Quizzes **Summative** Revision tests	
FEEDBACK SUPPORTS LEARNING	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓
SPECIALIST VOCABULARY	Bubble sort Merge sort Trace Flowcharts Linear search Binary search	Functions Procedures Iteration Flowcharts Pseudo code Test tables	Functions Procedures Iteration Flowcharts Pseudo code Test tables	See year 10 Curriculum map for keywords related to each topic.	See year 10 Curriculum map for keywords related to each topic.	
QUALITY FIRST TEACHING	✓ Strategies to learn more, remember more (metacognition) used in lessons e.g. retrieval, elaboration, interleaving, dual coding, etc. ✓ Differentiation and reasonable adjustments for students with SEND, EAL, etc. such as scaffolding, visual aids, audio, physical resources, planned questioning, etc. ✓ Opportunities for Literacy, Numeracy and Oracy, including a focus on reading ✓ Opportunities to apply key concepts and address misconceptions					

Rutlish School

YEAR 12						
	Autumn 1	**Autumn 2**	**Spring 1**	**Spring 2**	**Summer 1**	**Summer 2**
KNOWLEDGE	**UNIT OF WORK: The Characteristics of Contemporary Processors, Input, Output and Storage Devices**	**UNIT OF WORK: Data Types**	**UNIT OF WORK: Assembly Language – Software Development**	**UNIT OF WORK: Exchanging Data**	**UNIT OF WORK: Legal, Moral, Cultural and Ethical Issues**	**UNIT OF WORK: Algorithms**
	Recall and understand the different components of the CPU and its function. (Recall knowledge from KS 4). Describe the FDE cycle. Explore the factors that affect the performance of the CPU. Know and understand the busses in the CPU and the function of each bus.	Recall subject knowledge from KS 4. Develop knowledge and understanding of: Binary vs denary Hex, ASCII and Unicode, Utilities, function of the OS. Recall knowledge on Binary addition and subtraction. Introduce assembly language and understand the need for a variety of programming paradigms.	Recall subject knowledge from KS 4. Develop knowledge and understanding of: high level and low level programming languages. Introduce assembly language and understand the need for a variety of programming paradigms. Students will know and understand how to build a basic assembly language program.	Recall subject knowledge from KS 4. Develop knowledge and understanding of: What network protocols are and why are they needed. Re-visit subject knowledge on the TCP/IP stack and be able to describe the layers of the TCP/IP stack. Develop knowledge and understanding of network security and threats, network hardware and client server and peer-to-peer networks.	Students become aware of computing legislation such as the DPA, CMA, CDPA and Regulation of Investigatory powers. Students build their knowledge of case studies related to these legislation and apply their knowledge on different scenarios.	Recall subject knowledge from KS 4. Develop knowledge and understanding of: Searching algorithms, bubble sort and merge sort. Explore how these algorithms are represented in pseudo code and as programs.
KEY SKILLS	Be able to recall, understand and apply theoretical knowledge in the following ways: Compare CISC vs Risc Contrast Evaluate – performance of CPU Explain - FDE Describe – functions of processor components Contrast the difference between CISC and RISC processors, Ram vs Rom. Explain virtual storage.	Be able to recall, understand and apply theoretical knowledge in the following ways: Convert binary and denary numbers. Convert Hex to binary and denary. Add and subtract binary numbers Convert floating point and fixed point binary numbers. Perform bitwise manipulation and masks. Convert sign and magnitude and Two's complement binary numbers. Develop understanding and be able to explain utilities and the function of the OS.	Develop programming skills in assembly language. Be able to read a program in assembly language and trace the output of the program. Be able to link a program in assembly language to the memory locations data is saved in.	Be able to recall, understand and apply theoretical knowledge in the following ways: Explain the characteristics of a network and evaluate the different types of networks. Explain the internet structure. Explain, analyse and evaluate network threats. Understand and remember the difference between CISC and RISC networks.	Be able to recall, understand and apply theoretical knowledge in the following ways: Analyse case studies and apply the correct legislation to a case study. Responding to long questions by applying Knowledge and Understanding, Application and Evaluation.	Code a bubble sort and merge sort algorithm. Write a bubble sort and merge sort algorithm in pseudo code.
HOW DO WE BUILD ON SKILLS AND KNOWLEDGE?	**KS4:** Skills and knowledge build on the KS 4 unit: Systems architecture. See KS 4 curriculum map. Further develop knowledge and understanding of the CPU and the components of the CPU.	**KS4:** Skills and knowledge build on the KS 4 unit Data representation. See KS 4 curriculum map. Further develop knowledge and understanding of binary, fixed point data representation, ASCII and Unicode, S & M and Two's Complement. Build on the theory knowledge of assembly language by coding a program in assembly language. Build on theory knowledge from KS 4: OS and Utilities.	**KS4:** Skills and knowledge build on the KS 4 unit computational thinkigng. See KS 4 curriculum map. Further develop knowledge and understanding of low level programming languages i.e. assembly language. Build on the theory knowledge of assembly language by coding a program in assembly language.	**KS4** Skills and knowledge build on the KS 4 unit Networks and protocols. See KS 4 curriculum map. Further develop and build on knowledge and understanding of network topics such as: TCP/IP Network topologies, network hardware and network security.	**KS4:** Skills and knowledge build on the KS 4 unit Ethical, legal and cultural. See KS 4 curriculum map. Further develop and build on knowledge and understanding of legislation such as: DPA, CMA, CDPA, ethical and cultural issues.	**KS4:** Skills and knowledge build on the KS 4 computational units - See KS 4 curriculum map. Students extend their knowledge on KS 4 by being able to code the algorithms as well.

LINKS TO THE WORLD i.e. links to careers; equality: gender, class, ethnicity, etc.; different subjects	**Careers:** Link to Jon Von Neuman - Computer scientist & developer of Von Neuman architecture.	**Careers:** Alan Turing – equalities - see ppt.	**Careers:** See ppt slide on links to careers in programming. **Cross curricula** Link to Maths – numeracy.	**Careers:** See ppt slide on links to careers in networking jobs. **Cross curricular** Link to Maths – numeracy. Link to English - extended writing questions.	Links to ethics and cultural differences e.g. meaning of colours in different countries and how it relates to web design. Edward Snowdon case study, AI. **Cross curricular** DT - Graphic design	**Careers:** See ppt slide on links to careers in programming. **Cross curricular** Link to Maths – numeracy.
ASSESSMENTS Summative and Formative as applicable	**Formative:** Self and peer assessment Class discussions Classwork and homework Exit tickets Think pair share Quizzes Mnemonics Flipped learning **Summative assessment:** End of topic test	**Formative:** Self and peer assessment Class discussions Classwork and homework Exit tickets Think pair share Quizzes Mnemonics Flipped learning Pair programming and feedback. **Summative assessment:** End of topic test	**Formative:** Self and peer assessment Class discussions Classwork and homework Exit tickets Think pair share Quizzes Mnemonics **Summative assessment:** End of topic test	**Formative:** Self and peer assessment Class discussions Classwork and homework Exit tickets Think pair share Quizzes Mnemonics Flipped learning **Summative assessment:** End of topic test	**Formative:** Self and peer assessment Class discussions Classwork and homework Exit tickets Think pair share Quizzes Mnemonics Flipped learning **Summative assessment:** End of topic test	**Formative:** Self and peer assessment Class discussions Classwork and homework Exit tickets Think pair share Quizzes Mnemonics Flipped learning **Summative assessment:** End of topic test
FEEDBACK SUPPORTS LEARNING	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓
SPECIALIST VOCABULARY	ALU CU FDE CPU Von Neuman Multicore and Parallel systems Ram /Rom Virtual storage CISC /RISC Pipelining	Binary Hex Floating point / Fixed point Normalisation Sign and magnitude Two's complement ASCII, Unicode Bitwise manipulation	Assembly language Mnemonic INP STA BRA DAT HLT SUB, MUL, ADD BRP,BRZ	TCP/IP Firewall Routers, hub Packet switching/ Circuit switching Mac address Worms, Trojans, viruses Packet filtering	DPA, SMA CDPA Legislation Privacy Censorship	Merge sort Bubble sort Decomposition Flag Iteration Selection
QUALITY FIRST TEACHING	✓ Strategies to learn more, remember more (metacognition) used in lessons e.g. retrieval, elaboration, interleaving, dual coding, etc. ✓ Differentiation and reasonable adjustments for students with SEND, EAL, etc. such as scaffolding, visual aids, audio, physical resources, planned questioning, etc. ✓ Opportunities for Literacy, Numeracy and Oracy, including a focus on reading ✓ Opportunities to apply key concepts and address misconceptions					

Rutlish School

YEAR 12: COURSEWORK						
	Autumn 1	**Autumn 2**	**Spring 1**	**Spring 2**	**Summer 1**	**Summer 2**
KNOWLEDGE	**UNIT OF WORK:** **2.2.1 Programming techniques**	**UNIT OF WORK:** **2.2.1 Programming techniques**	**UNIT OF WORK:** **2.2.1 Programming techniques and 1.3.4 Web Technologies and NEA**	**UNIT OF WORK:** **2.3.1 Algorithms and NEA**	**UNIT OF WORK:** **2.3.1 Algorithms and NEA**	**UNIT OF WORK:** **2.3.1 Algorithms and NEA**
	Recall subject knowledge from KS 4. Develop knowledge and understanding of: Programming constructs: sequence, iteration, branching. how it can be used and compares to an iterative approach. Global and local variables. Modularity, functions and procedures, parameter passing by value and by reference. Use of an IDE to develop/debug a program. Develop knowledge and understanding of: Recursion and use of object oriented techniques. Create programs using programming techniques including OOP.	Develop knowledge and understanding of object oriented techniques. Create programs using programming techniques including OOP.	Develop knowledge and understanding of object oriented techniques. Create programs using programming techniques including OOP. Recall subject knowledge from KS 3: HTML, CSS Develop knowledge and understanding of: JavaScript. Create a website using HTML, CSS and JS NEA Overview: Learners will be expected to analyse, design, develop, test, evaluate and document a program written in a suitable programming language. The underlying approach to the project is to apply the principles of computational thinking to a practical coding problem. Learners are expected to apply appropriate principles from an agile development approach to the project development NEA: Analysis of the problem Describe and justify the features that make the problem solvable by computational methods. Explain why the problem is amenable to a computational approach.	Analysis and design of algorithms for a given situation. The suitability of different algorithms for a given task and data set, in terms of execution time and space. Measures and methods to determine the efficiency of different algorithms, Big O notation (constant, linear, polynomial, exponential and logarithmic complexity). Comparison of the complexity of algorithms. Recall subject knowledge from KS 4 of standard algorithms: bubble sort, insertion sort, merge sort, binary search and linear search Develop knowledge and understanding of: quick sort Create programs using programming techniques including OOP. NEA: Design of the solution Break down the problem into smaller parts suitable for computational solutions justifying any decisions made.	Develop knowledge and understanding of algorithms for the main data structures, (stacks, queues, trees, linked lists, depth-first (post-order) and breadth-first traversal of trees). Create programs using programming techniques including OOP. NEA: Design of the solution Break down the problem into smaller parts suitable for computational solutions justifying any decisions made. Explain and justify the structure of the solution. Describe the parts of the solution using algorithms justifying how these algorithms form a complete solution to the problem. Describe usability features to be included in the solution. Identify key variables / data structures / classes justifying choices and any necessary validation. Identify the test data to be used during the iterative development and post development phases and justify the choice of this test data.	Develop knowledge and understanding of standard algorithms: Dijkstra's shortest path algorithm, A* algorithm, Create programs using programming techniques including OOP. NEA: Developing the solution. Iterative development process and testing to inform development Provide annotated evidence of each stage of the iterative development process justifying any decision made. Provide annotated evidence of prototype solutions justifying any decision made. Provide annotated evidence for testing at each stage justifying the reason for the test. Provide annotated evidence of any remedial actions taken justifying the decision made.

			Identify and describe those who will have an interest in the solution explaining how the solution is appropriate to their needs (this may be named individuals, groups or persona that describes the target end user). Research the problem and solutions to similar problems to identify and justify suitable approaches to a solution. Describe the essential features of a computational solution explaining these choices. Explain the limitations of the proposed solution. Specify and justify the solution requirements including hardware and software configuration (if appropriate). Identify and justify measurable success criteria for the proposed solution.	Explain and justify the structure of the solution. Describe the parts of the solution using algorithms justifying how these algorithms form a complete solution to the problem. Describe usability features to be included in the solution. Identify key variables / data structures / classes justifying choices and any necessary validation. Identify the test data to be used during the iterative development and post development phases and justify the choice of this test data.		
KEY SKILLS	Be able to recall, understand and apply theoretical knowledge in the following ways: **Use of an IDE** **Practical skills in:** Basic programming concepts such as assignment and the use of math operators Selection Iteration Functions Recursion OOP Python Problem solving, debugging and critical thinking	Be able to recall, understand and apply theoretical knowledge in the following ways: **Use of an IDE** **Practical skills in:** Basic programming concepts such as assignment and the use of math operators Selection Iteration Functions Recursion OOP Python Problem solving, debugging and critical thinking	Be able to recall, understand and apply theoretical knowledge in the following ways: **Use of an IDE** **Practical skills in:** Basic programming concepts such as assignment and the use of math operators Selection Iteration Functions Recursion OOP Python HTML CSS JS Problem solving, debugging and critical thinking	Be able to recall, understand and apply theoretical knowledge in the following ways: **Use of an IDE** **Practical skills in:** Basic programming concepts such as assignment and the use of math operators Selection Iteration Functions Recursion OOP Programming of and implementation of algorithms Problem solving, debugging and critical thinking	Be able to recall, understand and apply theoretical knowledge in the following ways: **Use of an IDE** **Practical skills in:** Basic programming concepts such as assignment and the use of math operators Selection Iteration Functions Recursion OOP Programming of and implementation of algorithms Problem solving, debugging and critical thinking	Be able to recall, understand and apply theoretical knowledge in the following ways: **Use of an IDE** **Practical skills in:** Basic programming concepts such as assignment and the use of math operators Selection Iteration Functions Recursion OOP Programming of and implementation of algorithms Problem solving, debugging and critical thinking
HOW DO WE BUILD ON SKILLS AND KNOWLEDGE?	Skills and knowledge build on the KS 4 computational units – 2.2 Programming techniques. See KS 4 curriculum map. Students extend their knowledge on KS 4 by programming more complex programs and learning recursion and OOP	Skills and knowledge build on the KS 4 computational units – 2.2 Programming techniques. See KS 4 curriculum map. Students extend their knowledge on KS 4 by programming more complex programs and learning recursion and OOP	Skills and knowledge build on the KS 4 computational units – 2.2 Programming techniques. See KS 4 curriculum map. Students extend their knowledge on KS 4 by programming more complex programs and learning recursion and OOP. They also build knowledge from KS3 Year 8 HTML unit. Builds on GCSE CS NEA	Skills and knowledge build on the KS 4 computational units - See KS 4 curriculum map. Students extend their knowledge on KS 4 by being able to code the algorithms as well. Builds on GCSE CS NEA	Skills and knowledge build on the KS 4 computational units - See KS 4 curriculum map. Students extend their knowledge on KS 4 by being able to code the algorithms as well. Builds on GCSE CS NEA	Skills and knowledge build on the KS 4 computational units - See KS 4 curriculum map. Students extend their knowledge on KS 4 by being able to code the algorithms as well. Builds on GCSE CS NEA

Rutlish School

LINKS TO THE WORLD i.e. links to careers; equality: gender, class, ethnicity, etc.; different subjects	**Careers:** Programmer, software engineer, general IT **Cross curricular** Link to Maths – numeracy and algebra. Link to English - extended writing questions. NEA documentation.	**Careers:** Programmer, software engineer, general IT **Cross curricular** Link to Maths – numeracy and algebra. Link to English - extended writing questions. NEA documentation.	**Careers:** Programmer, software engineer, general IT **Cross curricular** Link to Maths – numeracy and algebra. Link to English - extended writing questions. NEA documentation. Links to DT – the design process and iterative design	**Careers:** Programmer, software engineer, general IT **Cross curricular** Link to Maths – numeracy and algebra. Link to English - extended writing questions. NEA documentation. Links to DT – the design process and iterative design	**Careers:** Programmer, software engineer, general IT **Cross curricular** Link to Maths – numeracy and algebra. Link to English - extended writing questions. NEA documentation. Links to DT – the design process and iterative design	**Careers:** Programmer, software engineer, data scientist, general IT **Cross curricular** Link to Maths – numeracy and algebra. Link to English - extended writing questions. NEA documentation. Links to DT – the design process and iterative design
ASSESSMENTS Summative and Formative as applicable	**Formative:** Self and peer assessment Class discussions Classwork and homework Exit tickets Think pair share Quizzes Mnemonics Flipped learning Pair programming and feedback. **Summative assessment:** Mid topic test	**Formative:** Self and peer assessment Class discussions Classwork and homework Exit tickets Think pair share Quizzes Mnemonics Flipped learning Pair programming and feedback. **Summative assessment:** End of topic test	**Formative:** Self and peer assessment Class discussions Classwork and homework Exit tickets Think pair share Quizzes Mnemonics Flipped learning Pair programming and feedback. **Summative assessment:** End of topic test	**Formative:** Self and peer assessment Class discussions Classwork and homework Exit tickets Think pair share Quizzes Mnemonics Flipped learning Pair programming and feedback.	**Formative:** Self and peer assessment Class discussions Classwork and homework Exit tickets Think pair share Quizzes Mnemonics Flipped learning Pair programming and feedback.	**Formative:** Self and peer assessment Class discussions Classwork and homework Exit tickets Think pair share Quizzes Mnemonics Flipped learning Pair programming and feedback. **Summative assessment:** Year 12 Mock
FEEDBACK SUPPORTS LEARNING	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓
SPECIALIST VOCABULARY	Python IDE Assignment Sequence Selection Iteration Functions Syntax/ Syntax Errors Logic errors Runtime errors	Object orientated programming Polymorphism Abstraction Inheritance Modularisation Scope Encapsulation Setters/Getters Instantiation Objects Methods Constructors Decorators	All Specialist Vocabulary from Autumn 2 HTML CSS Javascript	Big O Notation Linear Quadratic Polynomial Exponential Factorial Constant Logarithmic Binary Search Linear Search Bubble sort Insertion Sort Merge Sort	All Specialist Vocabulary from Spring 2 Quicksort Timsort Breadth graph traversal Depth graph traversal Nodes	All Specialist Vocabulary from Summer 1 A* Algorithm Dijkstra's Algorithm Travelling salesman problem Brute forcing
QUALITY FIRST TEACHING	✓ Strategies to learn more, remember more (metacognition) used in lessons e.g. retrieval, elaboration, interleaving, dual coding, etc. ✓ Differentiation and reasonable adjustments for students with SEND, EAL, etc. such as scaffolding, visual aids, audio, physical resources, planned questioning, etc. ✓ Opportunities for Literacy, Numeracy and Oracy, including a focus on reading ✓ Opportunities to apply key concepts and address misconceptions					

Rutlish School

YEAR 13						
	Autumn 1	**Autumn 2**	**Spring 1**	**Spring 2**	**Summer 1**	**Summer 2**
KNOWLEDGE	**UNIT OF WORK:** **NEA**	**UNIT OF WORK:** **Revision and NEA**	**UNIT OF WORK:** **Revision and NEA**	**UNIT OF WORK: Revision**	**UNIT OF WORK: Revision**	**UNIT OF WORK:**
	NEA: Developing the solution. Iterative development process and testing to inform development Provide annotated evidence of each stage of the iterative development process justifying any decision made. Provide annotated evidence of prototype solutions justifying any decision made. Provide annotated evidence for testing at each stage justifying the reason for the test. Provide annotated evidence of any remedial actions taken justifying the decision made. Create a game using programming techniques including OOP.	Recap subject knowledge for each topic using know more remember more strategies: Unit 5, 11 and 12 - Programming techniques, Algorithms, Web Technologies **NEA:** Developing the solution. Iterative development process and testing to inform development Provide annotated evidence of each stage of the iterative development process justifying any decision made. Provide annotated evidence of prototype solutions justifying any decision made. Provide annotated evidence for testing at each stage justifying the reason for the test. Provide annotated evidence of any remedial actions taken justifying the decision made. Create a game using programming techniques including OOP.	Recap subject knowledge for each topic using know more remember more strategies: Unit 5, 11 and 12 - Programming techniques, Algorithms, Web Technologies **NEA:** Evaluation Provide annotated evidence of testing the solution of robustness at the end of the development process. Provide annotated evidence of usability testing (user feedback) Use the test evidence from the development and post development process to evaluate the solution against the success criteria from the analysis Provide annotated evidence of the usability features from the design, commenting on their effectiveness. Discuss the maintainability of the solution. Discuss potential further development of the solution.	Recap subject knowledge for each topic using know more remember more strategies: Unit 5, 11 and 12 - Programming techniques, Algorithms, Web Technologies	Recap subject knowledge for each topic using know more remember more strategies: Unit 5, 11 and 12 - Programming techniques, Algorithms, Web Technologies	End of course.
KEY SKILLS	Be able to recall, understand and apply theoretical knowledge in the following ways: Reading and understanding code in Python Use of an IDE Practical skills in: Basic programming concepts such as assignment and the use of math operators, Selection, Iteration, Functions, Recursion OOP, Programming of and implementation of algorithms Problem solving, debugging and critical thinking	Be able to recall, understand and apply theoretical knowledge in the following ways: Reading and understanding code in Python Use of an IDE Practical skills in: Basic programming concepts such as assignment and the use of math operators, Selection, Iteration, Functions, Recursion OOP, Programming of and implementation of algorithms Problem solving, debugging and critical thinking	Be able to recall, understand and apply theoretical knowledge in the following ways: Understand and remember the key facts. Exam technique – respond to long questions: Knowledge and understanding, application and evaluation. Analyse and apply knowledge to given scenarios.	Be able to recall, understand and apply theoretical knowledge in the following ways: Understand and remember the key facts. Exam technique – respond to long questions: Knowledge and understanding, application and evaluation. Analyse and apply knowledge to given scenarios.	Be able to recall, understand and apply theoretical knowledge in the following ways: Understand and remember the key facts. Exam technique – respond to long questions: Knowledge and understanding, application and evaluation. Analyse and apply knowledge to given scenarios.	

HOW DO WE BUILD ON SKILLS AND KNOWLEDGE?	**Revisit subject knowledge:** Builds on GCSE CS NEA	**Revisit subject knowledge:** Builds on GCSE CS NEA Practice past papers Complete revision exercise C & D quizzes Revision worksheets	**Revisit subject knowledge:** Builds on GCSE CS NEA Practice past papers Complete revision exercise C & D quizzes Revision worksheets	**Revisit subject knowledge:** Practice past papers Complete revision exercise C & D quizzes Revision worksheets	**Revisit subject knowledge:** Practice past papers Complete revision exercise C & D quizzes Revision worksheets	
LINKS TO THE WORLD **i.e. links to careers; equality: gender, class, ethnicity, etc.; different subjects**	**Careers:** Programmer, software engineer, general IT **Cross curricular** Link to Maths – numeracy and algebra. Link to English - extended writing questions. NEA documentation. Links to DT – the design process and iterative design	**Careers:** Programmer, software engineer, general IT **Cross curricular** Link to Maths – numeracy and algebra. Link to English - extended writing questions. NEA documentation. Links to DT – the design process and iterative design	**Careers:** Programmer, software engineer, general IT **Cross curricular** Link to Maths – numeracy and algebra. Link to English - extended writing questions. NEA documentation. Links to DT – the design process and iterative design	**N/A** Covered in year 12	**N/A** Covered in year 12	
ASSESSMENTS **Summative and Formative as applicable**	**Formative:** Self and peer assessment Class discussions Classwork and homework Exit tickets Think pair share Quizzes Mnemonics Flipped learning Pair programming and feedback Major testing event: testing of NEA game with critical feedback from a wide range of KS3, 4 and 5 students	**Formative:** Use know more remember more strategies to revise content: Interleaving, spaced practice, retrieval practice, knowledge organisers, C & D quizzes. **Summative assessment:** Past paper practice.	**Formative:** Use know more remember more strategies to revise content: Interleaving, spaced practice, retrieval practice, knowledge organisers, C & D quizzes. **Summative assessment:** Past paper practice.	**Formative:** Use know more remember more strategies to revise content: Interleaving, spaced practice, retrieval practice, knowledge organisers, C & D quizzes. **Summative assessment:** Past paper practice.	**Formative:** Use know more remember more strategies to revise content: Interleaving, spaced practice, retrieval practice, knowledge organisers, C & D quizzes. **Summative assessment:** Past paper practice.	
FEEDBACK SUPPORTS LEARNING	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓
SPECIALIST VOCABULARY	See year 12 Curriculum map for keywords related to each topic.	See year 12 Curriculum map for keywords related to each topic.	See year 12 Curriculum map for keywords related to each topic.	See year 12 Curriculum map for keywords related to each topic.	See year 12 Curriculum map for keywords related to each topic.	
QUALITY FIRST TEACHING	✓ Strategies to learn more, remember more (metacognition) used in lessons e.g. retrieval, elaboration, interleaving, dual coding, etc. ✓ Differentiation and reasonable adjustments for students with SEND, EAL, etc. such as scaffolding, visual aids, audio, physical resources, planned questioning, etc. ✓ Opportunities for Literacy, Numeracy and Oracy, including a focus on reading ✓ Opportunities to apply key concepts and address misconceptions					

Rutlish School

YEAR 13	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
	UNIT OF WORK: Exchanging Data	**UNIT OF WORK: Revision**	**UNIT OF WORK: Revision**	**UNIT OF WORK: Revision**	**UNIT OF WORK: Revision**	**UNIT OF WORK:**
KNOWLEDGE	Recall subject knowledge from KS 4. Develop knowledge and understanding of: What network protocols are and why are they needed. Re-visit subject knowledge on the TCP/IP stack and be able to describe the layers of the TCP/IP stack. Develop knowledge and understanding of network security and threats, network hardware and client server and peer-to-peer networks.	Recap subject knowledge for each topic using know more remember more strategies: Develop and embed subject knowledge on processor components, processor performance, types of processors, I /O devices, storage devices, functions of OS, types of OS, programming language translators, programming paradigms, assembly language.	Recap subject knowledge for each topic using know more remember more strategies: Develop and embed subject knowledge on structure of internet, internet communication, network security and threats, html and css, client server and peer to peer, data types, binary, hex, Ascii, Unicode, binary arithmetic, floating point arithmetic, bitwise manipulation and masks.	Recap subject knowledge for each topic using know more remember more strategies: Computing related legislation, ethical, moral and cultural issues, privacy and censorship.	Recap subject knowledge for each topic using know more remember more strategies: Section 1- 9	End of course.
KEY SKILLS	Be able to recall, understand and apply theoretical knowledge in the following ways: Explain the characteristics of a network and evaluate the different types of networks. Explain the internet structure. Explain, analyse and evaluate network threats. Understand and remember the difference between CISC and RISC networks.	Be able to recall, understand and apply theoretical knowledge in the following ways: Understand and remember the key facts. Exam technique – respond to long questions: Knowledge and understanding, application and evaluation. Analyse and apply knowledge to given scenarios.	Be able to recall, understand and apply theoretical knowledge in the following ways: Understand and remember the key facts. Exam technique – respond to long questions: Knowledge and understanding, application and evaluation. Analyse and apply knowledge to given scenarios.	Be able to recall, understand and apply theoretical knowledge in the following ways: Understand and remember the key facts. Exam technique – respond to long questions: Knowledge and understanding, application and evaluation. Analyse and apply knowledge to given scenarios.	Be able to recall, understand and apply theoretical knowledge in the following ways: Understand and remember the key facts. Exam technique – respond to long questions: Knowledge and understanding, application and evaluation. Analyse and apply knowledge to given scenarios.	
HOW DO WE BUILD ON SKILLS AND KNOWLEDGE?	**KS4:** Skills and knowledge build on the KS 4 unit Networks and protocols. See KS 4 curriculum map. Further develop and build on knowledge and understanding of network topics such as: TCP/IP Network topologies, network hardware and network security.	**Revisit subject knowledge:** Practice past papers Complete revision exercise C & D quizzes Revision worksheets	**Revisit subject knowledge:** Practice past papers Complete revision exercise C & D quizzes Revision worksheets	**Revisit subject knowledge:** Practice past papers Complete revision exercise C & D quizzes Revision worksheets	**Revisit subject knowledge:** Practice past papers Complete revision exercise C & D quizzes Revision worksheets	
LINKS TO THE WORLD i.e. links to careers; equality; gender, class, ethnicity, etc.; different subjects	**Careers:** See ppt slide on links to careers in networking jobs. **Cross curricular** Link to Maths – numeracy. Link to English - extended writing questions.	Covered in year 12	Covered in year 12	Covered in year 12	Covered in year 12	

ASSESSMENTS Summative and Formative as applicable	**Formative:** Self and peer assessment Class discussions Classwork and homework Exit tickets Think pair share Quizzes Mnemonics Flipped learning **Summative assessment:** End of topic test	Use know more remember more strategies to revise content: Interleaving, spaced practice, retrieval practice, knowledge organisers, C & D quizzes. **Summative assessment:** Past paper practice.	Use know more remember more strategies to revise content: Interleaving, spaced practice, retrieval practice, knowledge organisers, C & D quizzes. **Summative assessment:** Past paper practice.	Use know more remember more strategies to revise content: Interleaving, spaced practice, retrieval practice, knowledge organisers, C & D quizzes. **Summative assessment:** Past paper practice.	Use know more remember more strategies to revise content: Interleaving, spaced practice, retrieval practice, knowledge organisers, C & D quizzes. **Summative assessment:** Past paper practice.	
FEEDBACK SUPPORTS LEARNING	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓	**Opportunity for students to reflect on learning, respond to feedback, improve work, etc.** ✓
SPECIALIST VOCABULARY	TCP/IP Firewall Routers, hub Packet switching/ Circuit switching Mac address Worms, Trojans, viruses Packet filtering	See year 12 Curriculum map for keywords related to each topic.	See year 12 Curriculum map for keywords related to each topic.	See year 12 Curriculum map for keywords related to each topic.	See year 12 Curriculum map for keywords related to each topic.	
QUALITY FIRST TEACHING	✓ Strategies to learn more, remember more (metacognition) used in lessons e.g. retrieval, elaboration, interleaving, dual coding, etc. ✓ Differentiation and reasonable adjustments for students with SEND, EAL, etc. such as scaffolding, visual aids, audio, physical resources, planned questioning, etc. ✓ Opportunities for Literacy, Numeracy and Oracy, including a focus on reading ✓ Opportunities to apply key concepts and address misconceptions					