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.

YEAR 7

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2	TEMP
	UNIT OF WORK: **Digital Literacy & Computer Systems**	**UNIT OF WORK:** **Data representation**	**UNIT OF WORK:** **Computational Thinking**	**UNIT OF WORK:** **Programming**	**UNIT OF WORK:** **Image Manipulation**	**UNIT OF WORK:** **Web Design**	
KNOWLEDGE	Understand how to use the learning platforms to be able to fully engage in all remote lessons, blended learning, how to access and submit homework and classwork set across the curriculum. Learn how to collaborate with other students safely and responsibly using the school remote learning sits. **E- Safety** To explore e-safety issues and understand how to prevent and protect the user using a range of digital devices. To learn about the different legislation regarding E-safety. Use technologies and online services securely, and know how to identify and report inappropriate conduct. Identify and explains how the use of technology can impact on society. **Introduction to computers** To look at the different parts of a computer system and understand what their function is. To learn about how a computer works and what is computer hardware and software. To learn about what a computer virus is and the different ways, it may catch a virus. To explore how to prevent a computer virus. To learn about how computers, collect data from other input devices and output data from different devices.	To explore the binary number system and to understand how the binary number system works. To learn what ASCII is and how ASCII is used to represent text and how ASCII is represented as binary. To learn about the different file sizes and how to calculate file sizes.	This is a practical unit covering the principles of producing control and monitoring solutions Using a flowchart-based interface (Flowol 4). To learn to produce systems that use simple loops and basic outputs, and then move on to look at systems that have multiple inputs and outputs. Students learn to refine their solutions using subroutines and variables. To research different types of computer and non-computerised control systems. To learn how to sequence instructions using the different flowchart symbols. Students will create a range of different flowcharts for different purposes. Understand what actuators are used for in control systems. Understand what a variable is and how they are used in a control system.	To learn about how to program IDE. To use the programming text based language python. To learn how to create basic programs using arithmetic operators, variables and inputs. To learn how to recognize and correct syntax errors in programs. To learn how to assign values to variables in programs and able to concatenate variables and string. To learn how predict outcomes of what will happen when a program runs.		To learn how to use web design software by creating a Multi-Page Website. To know and understand how to design a website that is fully fit for audience and purpose. To understand how the different features on a web page can be used to design an effective website. To understand the importance of copyright in web design. To be able to implement web components to make the website interactive.	
KEY SKILLS	**VLE** Logging on to Teams. Understand the difference between a task and an assignment. Be able to download and submit assignments; send a message to a teacher. **E – Safety** Understand how to be safe online and protect user's identities and personal information from any potentials dangers within the digital world of the Internet and **Electronic Devices**. Identify the differences with the following acts regarding the use of information. • The Data Protection Act	Converting from denary to binary and vice versa. Binary addition Binary subtraction ASCII representation Ordering binary units of measure, bit, and bytes. Calculating file sizes.	Use the function print to create a simple program. Use escape sequence \n to create a new line in a program. Use \t to create a tab space in a program. Create a program using commas to concatenate string and variables. Create programs to perform simple arithmetic calculations. Create a program that responds when you input personal details. Understand the different data types, string, integer, float and use it correctly in a program.	To plan the layout for a website to show what it will look like and that to explain how it will be suitable to the clients brief. To be able to use multi-media components (roll overs, hotspots, photo gallery, lightbox, search bar, time stamp, embed videos) effectively in a website. Use a Master page to ensure website design is consistent on all 3 pages. Able to create the Navigation system. To be able to create internal and external hyperlinks.	Identify everyday situations where computer control is used and the sensors used by control systems. Break down problems in the form of flowcharts. Design algorithms to solve problems by producing flowchart-based solutions for control systems that include sequences and loops, subroutines and variables. To be able to break a task down into its component steps and be able to correctly sequence the steps in a tasks Create a flowchart that shows all of the individual steps for a task,	To use code to program a triangle and square shape. Use a program to create students own shape. Use commands to move forward and backward. Use code to add different attributes to the Turtle. Use the begin_ fill() and end _fill () to get the program to fill a colour for different shapes. Use Use colour to change your back ground using the command bgcolor ("enter the color that you want) Use the command pensize(enter the size of the pen width) to change the pen width. Use commands for pen up and down.	

• Computer Misuse Act • Copyright Law **Intro to computers** Definition of software and hardware. The differences between hardware and software. Recognise the different types and the purpose of input and output devices How the computer can process data. Describe the purpose of the various computer components. Identify the types of computer viruses and how they occur. Understand how to prevent the risk of computer viruses.		To predict what a program will do and able to correct any syntax errors.	Use feedback to identify improvements and to be able to make appropriate refinements to a solution. To save and publish the website which loads quickly and correctly for all the webpages. To know how to include images on a webpage. Review and evaluate a website.	completely sequenced in the correct order. To create computer control systems that use subroutines Use Flowol for modelling traffic lights, temperature control and various other systems Create flow charts and computer control systems using the Flowol software package.	To use iteration, for loop and while loop in drawing different shapes.	
HOW DO WE BUILD ON SKILLS AND KNOWLEDGE? Draw upon student's knowledge and use of a VLE at Primary – make links to similarities of VLEs despite different applications used - see starter questions. Recap shortcuts and file types learned in primary school. **E – Safety** Review what students had been taught at Key Stage 2 using technology safely, respectfully and responsibly; identify ways to report concerns about the use of social media, online content or contact with other users. How in primary to search effectively and safely for information. Recognise how to avoid using inappropriate content and the skills in finding reliable data. **Intro to computers** Extend and expand upon the knowledge acquired in Key Stage 2, looking at what a digital computer is and the two different inter-related systems, hardware and software.	Draw upon knowledge of the decimal number line learned in KS 2 and make links to the binary number line – similarities and differences.	Draw upon the knowledge of using block-based Scratch programming in Key Stage 2 and develop working programs in Scratch to solve a range of problems. Understand the concepts of sequencing, selection and iteration. To compare and apply to learning Text-based Python programming.	Review what students had been taught at Key Stage 2 for Web Design. Connect with the rules and principles in creating suitable websites at Key Stage 2. Expand upon knowledge to explore the features of websites to learn how to create content for a web pages.	Review what students had been taught at Key Stage 2 Computing: Controlling Devices, programming and sequences using block based software.	Review work already covered in year 7 unit 2 Python programming autumn term 2. Identify key terms and definitions used in python programing.	
LINKS TO THE WORLD i.e. links to careers; equality: gender, class, ethnicity, etc.; different subjects **Cross curricular:** Students will be empowered to engage in the learning of all subjects as they can confidently use the learning platforms. Students will evaluate what they find online critically to be confident that the information is reliable and relevant to all subject areas. The rules for keeping safe online, how to recognise risks, harmful content and contact, and how to report them. **Careers:** Key skills link to world of work; students are able to use software, which is currently used in the world of work. Students are confident in being safe online and can effectively evaluate the credibility of information online acknowledge any risks, which will allow them to be responsible digital citizens in any future employment. Students can understand how a computer processes data and how the different components work to benefit current computer science, computer	**Cross curricular:** Understanding how base 2 links to numeracy skills (Maths). **Careers:** Software Developer. Database Administrator. Computer Hardware Engineer. Computer Systems Analyst. Computer Network Architect. Web Developer. Information Security Analyst. Computer and Information Research Scientists.	**Cross curricular:** Development of effective problem-solving skills in a way that is applicable to many other subjects and in many other areas of life. **Careers:** Students discovering their proficiency in problem solving, handling syntax, programming may find they can enter an industry in which those skills are highly valued. Computer Science industry: Robotics, British Aerospace, and Data Analysts.	**Cross curricular:** Development of effective research skills and evaluating the effectives of the content displayed on websites specific to all subject areas. **Careers:** Key skills to search effectively using websites for relevant information and data that can be applied to the following: Publishing, Media, Advertisement and Fashion industry. Web Science able to apply the skills to use the features to design an effective website.	**Cross curricular:** Development of problem-solving skills in a way that is applicable to many other subjects and in many other areas of life. **Careers:** Job opportunities: Software Programmers in: Games, Operating Systems, Databases, Application software, Mobile applications Air Traffic Control	**Cross curricular:** Development of problem-solving skills in a way that is applicable to many other subjects and in many other areas of life. **Careers:** Students discovering their proficiency in problem solving, handling syntax, programming may find they can enter an industry in which those skills are highly valued. Any careers linked to programming and different areas in Computer Science.	

	Col1	Col2	Col3	Col4	Col5	Col6	
	engineering or computer maintenance jobs. They are up to date with the risks in viruses to different computer digital devices used globally from the home environment to everyday work.						
ASSESSMENTS Summative and Formative as applicable	**Formative:** Class discussions – see ppt Quiz, Verbal feedback **Summative:** Task **E – Safety** **Formative:** Class discussions – see ppt Quiz What Well Well? (WWW) and Even Better If? (EBI) on project task. Mind map summarise what they already know about a topic in a creative and visual way **Summative:** End of unit task. End of unit quiz test. Self and peer evaluation of classwork. **Intro to computers** **Formative:** Class discussions – see ppt Quiz What Well Well? (WWW) and Even Better If? (EBI) on project task Mind map summarise what they already know about a topic in a creative and visual way **Summative:** End of unit Task End of unit quiz test Self and peer evaluation of classwork	**Formative:** Class discussions – see ppts and student workbook. Quiz on Fronter Spaced practice – recap of knowledge from previous lessons. Interleaving: recap of topic 1 – Introduction to computers. Self and peer evaluation of classwork. **Summative:** End of topic test.	**Formative:** Class discussions – see ppt Quiz What went well? (WWW) and Even Better If? (EBI) on project task Mind map summarise what they already know about a topic in a creative and visual way **Summative:** End of unit Task End of unit quiz test Self and peer evaluation of classwork.	**Formative:** Class discussions – see ppt Quiz on Fronter What went well? (WWW) and Even Better If? (EBI) on project task. Mind map summarise what they already know about a topic in a creative and visual way. Self and peer evaluation of classwork. **Summative:** End of unit Task End of unit quiz test.	**Formative:** Class discussions – see ppt Quiz What went well? (WWW) and Even Better If? (EBI) on project task Mind map summarise what they already know about a topic in a creative and visual way. Self and peer evaluation of classwork. **Summative** End of unit Task. End of unit quiz test.	**Formative:** Class discussions – see ppt. Quiz What went well? (WWW) and Even Better If? (EBI) on project task Mind map summarise what they already know about a topic in a creative and visual way. Self and peer evaluation of classwork. **Summative:** End of unit Task. End of unit quiz 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	VLE Download Upload Submit **E – Safety** Digital literacy Online safety Responsible use of technologies **Intro to computers** the physical components operating system Processor computer viruses	Bit Byte Kilobyte Megabyte Base 2 ASCII File size Data representation	Text based programming Block based programming Syntax Integer String Float/real Boolean Python Scratch Variable Input print test	Information Audience World Wide Web Published Web Components Evaluation Master page Hyperlinks Test Plan Effective search	Variables Model Flowcharts Evaluation Sequencing Input Subroutines Sensors Control Algorithm Process Output Decision Variable Mimic Actuators (Car Park control systems Computer Programming •	Python bgcolor Algorithm command String pensize Float/ Real ITERATION Character FOR LOOP Boolean WHILE LOOP Integer KEY PRESSES Syntax LISTENERS Variable Commenting CLASS OOP ENCAPSULATION ATTRIBUTE begin_fill() end_fill()	
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 8

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2	TEMP
KNOWLEDGE	**UNIT OF WORK:** **DATA REPRESENTATION** To understand the importance of file / Folder Management/ Passwords / Security. To understand the importance of online safety and how to stay safe online. **HTML E SAFETY WEBSITE**　　To learn the basics of HTML and CSS. To learn to create web pages using HTML. ＿＿＿＿＿＿＿To learn how to create an effective E-safety Web site design which adapts to any size of screen for viewing on, say, a mobile phone or a PC.＿＿＿＿To learn how to use a range of HTML tags to create well laid out website. **Set 4 & Set 5** **E-Safety Web design** To learn how to use Web design software to create a Multi-Page Website. To know and understand how to ensure that a website is fully fit for audience and purpose. To understand the importance of planning a website that meets the requirements of the brief.	**UNIT OF WORK:** **PROGRAMMING** To learn about the different units of data that can be stored inside a computer and how quantities of these binary digits represent all forms of data within a computer. To learn to convert denary numbers into binary and calculate the sum of binary addition calculations and binary subtraction calculations. To learn and understand the Hexadecimal number system and convert a binary number to Hex and a Hex number to binary.	**UNIT OF WORK:** **NETWORKS AND THE WEB** To learn more complex programming techniques using a text based language. Students will review and develop their knowledge of the programming constructs learned in year 7. Students will learn about the following: Print Loops Variables If Statements Else Statements To understand the use of arithmetic expressions. To read and create simple programs that use selection. To review Different data types, string, integer, float and Boolean. Know about different relational operators to form logical expressions. To explore what Boolean operators are. To predict answers to conditions and test out in python. Using flowcharts to write an algorithm for coding tasks. Describe how iteration (while statements) controls the flow of program execution. To learn how to use a range of loops in python programming: Count Controlled loop and while loops. To understand how to debug coding errors - Identify and fix common syntax errors. To develop knowledge of programming constructs to be able to predict outcomes of what will happen when a program runs.	**UNIT OF WORK :** **IMAGE EDITING & WEB DESIGN** In this unit, students will be able to use graphic editing software to create and edit graphics To learn how to use a wide range of Photoshop tools to create and manipulate graphic images. To understand the legislation regarding copyright issues. To use simple tools to edit/distort images and saving images in the correct locations. To recall tools used and explain how to use them. To learn the key concept of "Cloning" & how to "remove" unwanted aspects from an image by cloning another part of an image in its place. To learn the key concept of removing backgrounds and why this is often necessary when making attractive documents. To understand the purpose of different file types and their properties.	**UNIT OF WORK:** **SPREADSHEETS** To learn what a computer network is and will look at the advantages and disadvantages of using standalone vs networked devices. Students will know about different types of networks including LAN's and WAN's before looking at specific network topologies. Students will gain an understanding of key networking terminology and will be able to describe what the internet is, how it functions and how it differs from the World Wide Web. This unit progresses students' knowledge and understanding of networks and associated hardware. The unit will equip students to establish a foundation understanding of how data is transmitted across networks, as well as exploring the factors that can affect performance. Develop knowledge about the internet and services provided over the internet.	**UNIT OF WORK:** **COMPUTER CRIMES & LEGISLATION** To learn and understand the fundamentals of how a computer system operates. To learn about the hardware, software and memory of a computer system. To learn about the history and development of computing. Set 4 & Set 5 Animation To discuss the impact of 3D animation on the wider world. To know and understand the differences between key framing and stop motion animation. To undertake a creative project that involves using industry standard software Adobe Animate CC. To achieve challenging goals, including selecting relevant information, advertisement persuasive captions and meeting the needs of known users.	
KEY SKILLS	**Students will be able to explain:** • How to use school network appropriately and efficiently? • How to save work • To be able to contribute to class discussions e.g.: What are the benefits of sharing information online? **HTML E SAFETY WEBSITE** To create text styles and add content, including text and graphics, in a specified position on a page, as well as navigation links to other pages on their website and to external websites. Explain some good and not so good features on a web page Demonstrate you can add features using HTML coding. Explain the term HTML. Demonstrate you can add HTML code to a web page. Demonstrate you can	**Students will be able to explain:** Why data must be stored in binary format, converted to be processed by a computer. Be able to explain the data units and moving between each – Bit, Nibble, Byte, Kilobyte, Megabyte, and Gigabyte. **Students will be able to do the following:** Convert positive denary whole numbers to binary numbers (up to and including 8 bits) and vice versa. Addition and subtraction – two binary integers (up to and including 8 bits) and explain overflow errors which may occur. Convert positive denary whole numbers into 2-digit hexadecimal numbers and vice versa.	To use simple arithmetic expressions in assignment statements to calculate values. Receive input from the keyboard and convert it to a numerical value. To write programs that display messages, receive keyboard input, and use simple arithmetic expressions in assignment statements. To use selection (if-elif-else statements) to control the flow of program execution. To locate and correct common syntax errors. I can create lists and access individual list items. To use flowcharts to write an algorithm for grading and calculator program. To write a program in python to check the number within acceptable limits and outputs a message saying accepted or not. To define and use iteration in a program. Combine iteration and selection to control the flow of program execution. To confidently modify and write a program using a while loop and a for loop. To be able to explain and predict outcomes	Learn to use graphic editing software to design and manipulate images to create a movie poster using a suitable graphics package in this unit. To learn skills in design, photo editing and image manipulation to create graphics for a given audience and purpose. To create a movie poster using the skills they have learnt so far. To apply different formatting techniques and discuss copyright law. Select appropriate images for the project brief. Demonstrate an understanding of licensing issues involving online content by applying for proper Creative Commons licences. Demonstrate the ability to credit the source of an image. To use the distort features within the software to produce practical effects.	Pupils will learn that the World Wide Web is part of the Internet, and how web addresses are constructed and stored. Understand and explain what an IP address is, Client-server, peer-to-peer networks. Understand and explain the concept of cloud Computing. Understand and explain ways of keeping data secure and simple encryption techniques. To understand how data is sent across a network. Know the role of a range of basic hardware involved in networking, such as switches Understand domain names and DNS. Know about a range of Internet services.	Identify key people and explain what they did to help in the development of computers. Identifying computer components. Describe the input, process and output cycle for four standard procedures. Explain the purpose of ROM and RAM in a computer. To explain how each computer part affects the performance of the computer system. To create a presentation to show a computer works. **Animation** To add, delete, and move objects/images. To scale and rotate objects. To be able to add colour to objects/images.	

accurately source images for your web site. Modify HTML tags using inline styling to improve the appearance of web pages. Display images within a web page. Apply HTML tags to construct a web page structure from a provided design. To experiment with CSS to format tags in a HTML document - Describe what CSS is - Use CSS to style static web pages - Assess the benefits of using CSS to style pages instead of in-line formatting. **Set 4 & Set 5** **E-Safety Web design** To review an existing website and consider its structure. - I can explore a website - I can discuss the different types of media used on websites. - I know that websites are written in HTML To plan the features of a web page - I can recognise the common features of a web page - I can suggest media to include on my page relevant to E-Safety. - I can draw a web page layout that suits my purpose.To consider the ownership and use of images (copyright). - I can say why I should use copyright free images. - I can find copyright-free images. - I can describe what is meant by the term 'fair use'. To recognise the need to preview pages. - I can evaluate my web page. - I can explain what a navigation path is. - I can describe why navigation paths are useful - I can make multiple web pages and link them using hyperlinks. - I can evaluate the user experience of a Website.	Convert binary integers to their hexadecimal equivalents and vice versa. To perform binary shifts. To understand how characters are represented in binary.	of what will happen when a program runs. To be able to locate and correct common syntax errors.	To use the Blur features within the software to produce practical effects. To use the Pen features within the software to produce practical effects. To explain what cloning is using a photo-editing software package. To use the cloning tool to select a target background colour or texture. To adjust the brush to a suitable size for the selected image. To remove an object/person from an image so that it is hard to spot "with the naked eye". To understand how to delete a background using the magnetic lasso tool. Adjust their brush to a suitable size for the image and make a neat selection of the image. To paste a selected object onto a new background and position and edit if further. To export to PNG and Jpeg. To understand the correct file format to be exported suitable for poster and website format.	To know what a computer network is - Define what a computer network is and explain how data is transmitted between computers across networks - Define 'protocol' and provide examples of non-networking protocols Explain the meaning and significance of bandwidth. To compare wired and wireless technologies To know the difference between the internet and the World Wide Web. To discuss the relationship between IP address and domain name - Describe components (servers, browsers, pages, HTTP and HTTPS protocols, etc.) and how they work together.	To add, move, and delete keyframes to make basic animations. To play, pause, and move through the animation using the timeline. To understand how to create/source a wide range of assets; test that animation is functional and addresses the client brief; successfully uses tools and techniques to enhance the animation. To be able to: create a simple animation using simple drawing and frame-by-frame techniques. Explain how frame rate and speed affect the smoothness of the animation. Use multiple layers. Use tweening and frame-by-frame techniques. Add sound effects.	
HOW DO WE BUILD ON SKILLS AND KNOWLEDGE?	By using activities which favour different learning styles to generate creative / lateral thinking and solutions to problems posed by E Safety. Creating a nurturing environment in which to discuss / air concerns safely. **HTML E SAFETY WEBSITE** Students have built websites in Year 7 using web plus so this unit aims to expand those skills and to build upon them by developing knowledge in using HTML coding to build an E-Safety website. Students recognise how websites are put together and attempt to create their own pages using online HTML editors. **E-Safety Web design** **Set 4 & Set 5** Students have built websites in Year 7 using web plus so this unit aims to	Draw upon knowledge of binary unit in year 7. Review and build upon knowledge gained in year 7 in how the binary number system works and converting binary numbers to denary. Learn about the Hexadecimal number system by firstly revisiting and revising their Binary knowledge from Year 7. Students will then learn about how to convert between Denary, Binary and Hexadecimal numbers.	Draw upon the knowledge of using Python programming in year 7 from the Python unit and python oop unit. Revisit the use of sequence, selection and repetition in programs; work with variables and various forms of input and output learn in year 7. To build on the skills of year 7 be developing the skills to recognize and correct syntax errors in programs. Variables, selection were learnt in year 7 and will be recapped this unit. New concepts such as iteration, random and lists are used to produce a more efficient quiz. All these skills are needed for GCSE Computer Science.	Continue to build on knowledge about copyright.	Review what students had been taught in key stage 2 and in unit 1 Introduction to computers understand computer networks including the internet; how they can provide multiple services such as the world wide web; and the opportunities they offer for communication and collaboration.	Review what students had been taught In key stage 2 and in year 7 introduction to computers unit. To continue to develop their understanding of how computer and technology works and how computers process instructions and commands **Set 4 & Set 5** **Animation** This links to KS2 by using animation skills taught during Year 4-6. This links to KS3 by using graphics skills taught during Year 8 in spring term.	

	expand those skills and to build upon them by developing knowledge in using advanced web components and a range of assets to build an E-Safety website.					

LINKS TO THE WORLD
i.e. links to careers; equality: gender, class, ethnicity, etc.; different subjects

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6
_HTML E SAFETY WEBSITE & Set 4 & Set 5 E-Safety Web design **Cross curricular:** Development of problem-solving skills in a way that is applicable to many other subjects and in many other areas of life. E-Safety is compulsory across all IT courses and links to the fundamentals of being safe online. **Careers:** Students learn by experience and grow as confident and responsible citizens who value and appreciate how webpages are built using HTML coding. They can adapt and transfer these creating and problem solving skills in other areas of the curriculum. They can aspire to be innovators and independent HTML coders that has an impact on careers in engineering, computer science and programming. **Job opportunities:** Web developer, web designer, front end developer. **Set 4 & Set 5 E-Safety Web design** **Cross curricular:** Development of effective research skills and evaluating the effectives of the content displayed on websites specific to all subject areas. **Careers:** Key skills to search effectively using websites for relevant information and data that can be applied to the following: In Publishing, Media, Advertisement and Fashion industry.	**Cross curricular:** Development of problem-solving skills in a way that is applicable to many other subjects and in many other areas of life. **Careers:** Students discovering their proficiency in problem solving, handling syntax, programming may find they can enter an industry in which those skills are highly valued. Computer Science industry: Robotics, British Aerospace, Data Analysts.	**Cross curricular:** Development of problem-solving skills in a way that is applicable to many other subjects and in many other areas of life. This unit links to the GCSE in computer science. The content in this unit will help towards component 2_Computational thinking, algorithms and programming. **Careers:** Students discovering their proficiency in problem solving, handling syntax, programming may find they can enter an industry in which those skills are highly valued. Computer Science industry: Robotics, British Aerospace, Data Analysts **Job opportunities:** Machine Learning Engineer, Python Developer, Data Scientist, DevOps Engineer, Full-Stack Developer	**Cross curricular:** DT- design work. Copyright and ownership - Students know that work they create belongs to them - They can name their work so that others know it belongs to them. When searching on the internet for content to use, students can explain why they need to consider who owns it and whether I have the right to reuse it. **Careers:** Key skills to search effectively using websites for relevant information and data that can be applied to the following: In Publishing, Media, Advertisement and Fashion industry.	**Cross curricular:** This unit links to the GCSE in computer science. The content in this unit will help towards component 1 - Computer Systems. The skills acquired in the use of research, describe and retain information can be applied to other subject areas. **Careers:** Network and Computer Systems Administrator. Computer and Information Systems Manager. Computer Network Architect. Computer Systems Analyst. Computer Network Support Specialist.	**Cross curricular:** Research, describe and retain information can be applied to all subject areas and is useful in higher education and most careers. **Set 4 & Set 5** **Animation** Understanding how to manipulate multimedia software across different platforms has become increasingly important in our digitally driven economy. It allows students to communicate ideas effectively, and these skills can be put to use across the curriculum Problem Solving Implementation, Evaluation and Analysis skills that can be applied to the world of work and other subject areas. **Careers:** Development of problem-solving skills in a way that is applicable to many other subjects and in many other areas of life. **Set 4 & Set 5 Animation** This links to careers by developing core skills needed roles in the computer animation and gaming industries. 2D animator 3D animator Background artist Character animator Clean-up artist Digital ink and paint artist Image editor Key frame animator

ASSESSMENTS Summative and Formative as applicable						
Formative: Class discussions – see ppt Quiz What went well and even Better If? (EBI) on project task Mind map summarise what they already know about a topic in a creative and visual way. Self and peer evaluation of classwork. **Summative:** End of unit Task End of unit quiz test To put evidence of their final website in an assessment.	**Formative:** Class discussions – see ppt Quiz What went well and even Better If? (EBI) on project task Mind map summarise what they already know about a topic in a creative and visual way. Self and peer evaluation of classwork. **Summative:** End of unit assessment. End of unit quiz test	**Formative:** Class discussions – see ppt Quiz What went well and even Better If? (EBI) on project task Mind map summarise what they already know about a topic in a creative and visual way. Self and peer evaluation of classwork. **Summative:** End of unit assessment. End of unit quiz test Create a series of programmes in response to set tasks.	**Formative:** Class discussions – see ppt Quiz What went well and even Better If? (EBI) on project task Mind map summarise what they already know about a topic in a creative and visual way. Self and peer evaluation of classwork. **Summative:** End of unit Task End of unit quiz test .	**Formative:** Class discussions – see ppt Quiz What went well and even Better If? (EBI) on project task Mind map summarise what they already know about a topic in a creative and visual way. Self and peer evaluation of classwork. **Summative:** End of unit Task End of unit quiz test. End of unit assessment.	**Formative:** Class discussions – see ppt Quiz What went well and even Better If? (EBI) on project task Mind map summarise what they already know about a topic in a creative and visual way. Self and peer evaluation of classwork. **Summative:** End of unit assessment. End of unit quiz test **Set 4 & Set 5** Create a short animation that will be evidenced in an Assessment Portfolio. They will also answer questions about their animations and complete a self-assessment.	

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						
HTML E SAFETY WEBSITE TAGS Test CSS HTM **Set 4 & Set 5** **E-safety** **Web design** Information Audience World Wide Web Published Web Components Evaluation Master page HyperlinksPlan Effective search Formatting a webpage	Hexadecimal Binary Denary Convert Base 2 Nibble, bit, byte, kilobyte, gigabyte	Problem Solving Programming Sequencing Loops If Statements De Bugging Selection Iteration	Image manipulation Photo editing Photoshop Magnetic lasso tool. Cloning Distort features Filters Layers Refine edge Quick Selection tool Magic wand tool Brush tool Healing tool Clipping mask	**Networks** HTTP and HTTPS Protocols, IP address and domain name hardware servers, browsers packet switching switch	Rom Ram Components Hardware multiple-core processors Input, process and output Motherboard Heat sink Speed (GHz) **Animate CC** workspace multiple layers tweening-Classic tween, motion tween, shape tween frame-by-frame techniques timeline persuasive captions elements	

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						✓

YEAR 9							
	Autumn 1	**Autumn 2**	**Spring 1**	**Spring 2**	**Summer 1**	**Summer 2**	**TEMP**
KNOWLEDGE	UNIT OF WORK: DATA REPRESENTATION	UNIT OF WORK: PROGRAMMING	UNIT OF WORK: STANDARD ALGORITHMS	UNIT OF WORK: IMAGE EDITING & MULTIMEDIA DESIGN	UNIT OF WORK: COMPUTATIONAL THINKING	UNIT OF WORK: WEB DESIGN	
	Knowledge: Create a sound understanding of how the Rutlish network works and how to remain safe online. An appreciation of how computers and networks function is crucial to understanding not only how to use them but to maintain them and remain safe whilst doing so. Growing social media use amongst teenagers and the attendant problems it brings is well documented and the curriculum intends to show pupils how to deal with difficult situations online and inform them of the people and places to whom they can turn. This unit of work should last between 5/6 weeks and so teachers should begin the Number Systems Unit before half term. The intention is to teach students about alternative number systems to the one we use	**Number Systems & Python** Number Systems continued from last term Python Programming The ability to use Python a high level programming language required for KS4 programmes of study. In so doing learners use computational thinking skills.	**Ethical and Legal issues** Students earn about ethical issues, environmental issues and legal issues with computing. They will take part in many discussions to look at how rapid technological changes are shaping people's perceptions, the environment and the laws that govern us all.	**Algorithms & Python** Students learn some of the different ways in which a computer can sort information. Students analyse the different methods to test for efficiency. Students will then perform the different methodologies using Python.	**Animation** Create a 60 second advert using Animate. Students use industry standard software to create advert for a product and go through different stages of planning, creating and testing a product, which they review at end.	**Creating a Multi-Page Website** Exploration of what makes a website good and bad and the opportunity to build a website using set criteria against which the final product can be evaluated.	
KEY SKILLS	• Understanding about different sources of problems online. • Analysing and evaluating the different agencies to whom they can turn. Also to develop oral and written comprehension skills. • Learn how to add / subtract in binary • Convert from binary to denary • Learn how to use Hex	**Number Systems Contd** Two's Complement ASCII Test **Python** If / Else Relational Operators Algorithms For Loops While Loops Test	To be able to describe and explain the term ethics; to list some of the issues that surround technology and its environment; to identify some of the issues in GDPR and to explain what is meant by the terms "fair use"; "file sharing" and "public domain".	Learning how to perform the following Bubble Sort Quick Sort Merge Sort	**Using** Key frames Tweening Onion skinning Free transform Adding sound	**Website** Using a master page Building minimum 3 pages Demonstration of following skills (not exhaustive list) • Inserting / editing navigation bar • Inserting header and footer • Inserting banner • Adding and editing appropriate text • Adding and editing images • Rollover Buttons • Photo Galleries • Hot Spots • Hyperlinks • Video links	

HOW DO WE BUILD ON SKILLS AND KNOWLEDGE?	• By using activities which favour different learning styles to generate creative / lateral thinking and solutions to problems posed by E Safety • Creating a nurturing environment in which to discuss / air concerns safely. • By developing skills required for number systems incrementally at KS3 level		The topic is introduced in year 8 but for year, 9 different scenarios are examined in order to consolidate prior learning but to stretch students further.	Topic introduced for the first time and is required for KS4 PoS	Introduced in Year 9 as prep for students taking IMedia in year 10	Students have built websites in Year 7 so this unit aims to consolidate those skills and to build upon them by adding a list of advanced interactive features for students to use.
LINKS TO THE WORLD i.e. links to careers; equality: gender, class, ethnicity, etc.; different subjects	➢ PSHE ➢ Potentially Drama ➢ Literacy – opportunities presenting by storyboarding ➢ Media – filming role play? ➢ Cyber Security in the real world ➢ Bletchley Park GCHQ		AI GDPR Computer Misuse Act	Just a thought - PE – if PE had a list of boys they needed to sort into teams / squads? Technology / IT	Animation / IT industry – Creative industries worth 12% UK GDP	PSHE – Animal Shelter – Looking after another living thing **Careers** Web Design Graphic Design Editorial Software Engineering
ASSESSMENTS Summative and Formative as applicable	**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 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 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	• Oversharing • Digital Footprint • Trolling • File management • Naming Conventions • Computer Network • Binary • Denary • Hex		Fair Use Public Domain File Sharing	Bubble Sort Merge Sort Quick Sort		
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

Rutlish School

YEAR 10: Computer Science: THEORY

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
	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
KNOWLEDGE	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

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 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	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
	UNIT OF WORK: <u>COMPUTATIONAL THINKING, ALGORITHMS AND PROGRAMMING</u>	**UNIT OF WORK:** <u>Algorithms & Pseudo Code</u>	**UNIT OF WORK:** <u>Subprograms</u>	**UNIT OF WORK:** <u>Subprograms</u>	**UNIT OF WORK:** <u>Lists and loops</u>	**UNIT OF WORK:** <u>Nested Lists & File Reading</u>
KNOWLEDGE	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	<u>**Use Python to apply the following practical skills: (Recall embed skills from KS 3)**</u> Escape sequences Code small programs Identify and correct syntax errors independently Maths operators Data type, variable and inputs String formatting Relational operators Selection	<u>**Use Python to apply the following practical skills:**</u> 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.	<u>**Use Python to apply the following practical skills:**</u> Decomposition Breaking larger programs down into subprograms. Difference between functions and procedures. Calling function/procedure Return a value to a function	<u>**Use Python to apply the following practical skills:**</u> Decomposition Breaking larger programs down into subprograms. Difference between functions and procedures. Calling function/procedure Return a value to a function	<u>**Use Python to apply the following practical skills:**</u> For loops While loops Built in functions Validation in programs Sub programs	<u>**Use Python to apply the following practical skills:**</u> Reading and writing to a file Nested lists Flowcharts Pseudo code
HOW DO WE BUILD ON SKILLS AND KNOWLEDGE?	<u>**KS 3 SOW**</u> Year 7 – Input, Output, variables, selection, data types	<u>**KS 3 SOW**</u> Practical programming using OR, AND to code programs making use of Boolean logic e.g. grade calculator, password program.	<u>**KS 3 SOW**</u> Year 9 SOW – selection and iteration.	<u>**KS 3 SOW**</u> Year 9 SOW – selection and iteration	<u>**KS 3 SOW**</u> Year 9 SOW – sorting and searching algorithms	<u>**KS 3 SOW**</u> Iteration Selection Decomposition

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	
	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
KNOWLEDGE	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.			

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 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	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

Rutlish School

YEAR 12

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
	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
KNOWLEDGE	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 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	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
UNIT OF WORK: **2.2.1 Programming techniques** **Software development**	**UNIT OF WORK:** **2.2.1 Programming techniques** **Computational thinking** **Into to NEA**	**UNIT OF WORK:** **2.2.1 Programming techniques and 1.3.4 Web Technologies and NEA** **Computational thinking** **Development of NEA**	**UNIT OF WORK:** **2.3.1 Algorithms and NEA** **Development of NEA**	**UNIT OF WORK:** **2.3.1 Algorithms and NEA** **Development of NEA**	**UNIT OF WORK:** **2.3.1** **Algorithms and NEA** **Development of NEA**

KNOWLEDGE

Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
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. Reading and writing Pseudo code. This will be introduced from the beginning using HTML, CSS and JS. Recall subject knowledge from KS 3: HTML, CSS Develop knowledge and understanding of: JavaScript. Create a website using HTML, CSS and JS Develop knowledge and understanding of: Recursion and use of object oriented techniques. Create programs using programming techniques including OOP. Students are introduced to aspects of software development: Waterfall, Agile and Spiral, Extreme and Rad. Students learn ow to write and follow algorithms. Students revise Pseudo code from KS 4 and develop their knowledge in read and writing pseudo code.	Develop knowledge and understanding of object oriented techniques. Create programs using programming techniques including OOP. Students are introduces to and will develop subject knowledge to: Understand the nature of and need for abstraction. Describe the differences between an abstraction and reality. Devise an abstract model for a variety of situations. Identify the inputs and outputs for a given situation. Determine the preconditions for devising a solution to a problem. Understand the need for reusable program components. Understand the nature, benefits and drawbacks of caching. Identify the components of a problem. Identify the components of a solution to a problem. Determine the order of the steps needed to solve a problem. Identify sub-procedures necessary to solve a problem.	Develop knowledge and understanding of object oriented techniques. Create programs using programming techniques including OOP. 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 the points where a decision has to be taken. Determine the logical conditions that affect the outcome of a decision. Determine how decisions affect program flow. Determine which parts of a program can be tackled at the same time. Determine the benefits and trade-offs that might result from concurrent processing in a particular situation.	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 Creating and solving algorithms.	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 Students develop their computational thinking 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 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. Students develop their skills by developing their iterative design prcess.	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 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	Python IDE Assignment Sequence Selection Iteration Functions Syntax/ Syntax Errors Logic errors Runtime errors Waterfall cycle Agile methodologies Extreme programming Spiral model Rapid application development	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					

YEAR 13

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
	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:**
KNOWLEDGE	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. Final submission of NEA prior to Feb half term.	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 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	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: <u>Exchanging Data</u>	**UNIT OF WORK:** Revision	**UNIT OF WORK:** <u>Revision</u>	**UNIT OF WORK:** <u>Revision</u>	**UNIT OF WORK:** <u>Revision</u>	**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 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	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					