SUBJECT Curriculum Map

Year 7

Carnforth High School

Rationale and Links to The National Curriculum
Computing skills are a vital component to any student's future success when they leave Carnforth High School. Through engaging content and a supportive environment, pupils will develop the skills and understanding that allow them to safely and confidently use technology in the real world; building their digital literacy, helping them become good "digital citizens" and preparing them for further study or job opportunities in the technology sector and beyond. At the heart of Computing is Computer Science, where students learn the key ideas behind information and computation, how digital systems function, and how to apply this knowledge through programming. With this foundation, they're able to confidently use information technology to design programs, build systems, and create all kinds of digital content. The design of the Key Stage 3 Computing curriculum revolves around ensuring pupils are equipped with the relevant skills that will enable them to successfully study either GCSE Computer Science or Creative iMedia as an option for Key Stage 4. It takes inspiration from a wide range of sources within the education sector, building on the experience of a number of professional bodies, such as the NCCE, PGOnline, CSUK and more, while focussing on providing content that is relatable to our pupils. The content in Year 7 allows pupils to take their first steps in meeting the skills and processes outlined in the Computing National Curriculum. Unit 1 revolves around understanding a range of ways to use technology safely, Unit 2 looks at understanding the hardware and software components that make up computer systems, Unit 3 and 4 give the opportunity to design, use and evaluate computational abstractions and introduces pupils to their first programming language (Scratch), Unit 5 and 6 allow pupils to undertake creative projects that involve selecting, using, and combining multiple applications, collecting and analysing data, create, re-use, revise and re-purpose digital artefacts.

	Half Term 1 – Digital Literacy (6 Lessons)	Half Term 2 – Computer Systems (7 Lessons)	Half Term 3 & Half Term 4 – Scratch Programming (11 Lessons)	Half Term 5 – Spreadsheets (6 Lessons)	Half Term 6 – Graphics Editing (6 Lessons)
Key Topics	• Logging in to and using school systems • Pupils know the use of social media • Protecting personal data	• What is a computer? • What is computer hardware and what are the main parts in a computer? • Using different input & output devices	• Planning, including the use of algorithms • Using block coding to create a programme • Sequencing instructions • Creating and using variables • Utilising user input • Using selection in a programme to make decisions	• How to add, edit and delete data in a spreadsheet • Effective formatting of a spreadsheet • Using mathematical operators to make calculations	• Understand bitmap and vector images • Sourcing assets and copyright law • Understand the properties of digital images and how they affect quality

	• Using Word and PowerPoint to present information • Using Outlook to manage emails • Using OneNote to share resources • Effective web searching	• Function of the CPU • What is Computer software and its different purposes? • Operating systems and their main functions	• Using Boolean operators in programming code • Testing a programme to make sure it meets the purpose	• Using formulae to carry out calculations automatically • Using functions, such as SUM, AVERAGE, MAX, MIN, COUNT, IF, and VLOOKUP • Presenting data in graphs	• Combine images and text in an impactful way • Use selection tools effectively • Be able to remove parts of an image • Use layers appropriately • Understand the use of different adjustment tools
Substantive Knowledge (The knowledge the students will develop)	• **School systems**: Pupils know login procedures, digital etiquette, and responsible use of school platforms. • **Social media**: Know how social media platforms work, including risks, benefits, and digital footprints. • **Data protection**: Pupils know what personal data is, why it needs to be protected, and how to stay safe online. • **Productivity tools**: Knowing the features and purposes of: a) **Microsoft Word** for document creation b) **PowerPoint** for presentation design c) **Outlook** for sending, receiving, and organising emails	• **Definition of a computer:** Pupils know what a computer is and its basic purpose. • **Computer hardware:** Identifying the physical components of a computer system, such as the CPU, RAM, storage, and peripheral devices. • **Input & output devices:** Recognising and describing various input (e.g., keyboard, mouse, scanner) and output devices (e.g., monitor, printer, speakers). • **Function of the CPU:** Pupils know the role of the Central Processing Unit as the brain of the computer, processing	• **Algorithms**: Pupils know what an algorithm is and how it can be used to plan a sequence of steps to solve a problem. • **Block-based coding**: Knowing how Scratch uses visual blocks to represent code and build programs. • **Sequencing**: Pupils know that the order of instructions affects the outcome of a program. • **Variables**: Knowing what variables are and how they store data within a program. • **User input**: Pupils know how a program can take input from a user to affect its behaviour. • **Selection (if/else)**: Pupils know how a program can make decisions using conditional statements. • **Boolean logic**: Recognising and applying Boolean operators (e.g., and, or, not) within conditions. • **Testing and debugging**: Knowing the importance of testing a program to check if it works correctly and how to find and fix errors.	• **Basic spreadsheet operations**: Pupils know how to add, edit, and delete data in spreadsheet cells. • **Formatting techniques**: Knowing how to format cells effectively (e.g., fonts, colours, borders, number formats) to improve readability and presentation. • **Mathematical operators**: Recognising and using basic operators like +, –, ×, ÷ within formulas. • **Formulas and calculations**: Pupils know how to create and use formulas for automatic calculations. • **Common functions**: Knowing the purpose and use of built-in	• **Bitmap vs Vector images**: Pupils know the differences between these image types (e.g., scalability, file size, resolution). • **Digital image properties**: Knowing how resolution, dimensions, and file formats affect image quality and performance. • **Sourcing digital assets legally**: Pupils know copyright laws, licensing (e.g., Creative Commons), and ethical use of images. • **Image composition**: Knowing how to combine text and images effectively to create visually appealing and purposeful designs. • **Adjustment tools**: Pupils know tools

	d) **OneNote** for digital note-taking and collaboration • **Web searching**: Pupils know how search engines work and how to search effectively using keywords and operators.	instructions and managing tasks. • **Computer software:** Knowing what software is, including the distinction between application software and system software. • **Operating systems:** Pupils know what an operating system does, including managing hardware, running applications, and providing a user interface.		spreadsheet functions, such as: a) SUM, AVERAGE, MAX, MIN, COUNT: for basic data analysis b) IF: for conditional logic • **Data visualisation**: Pupils know how to present data effectively using different types of graphs (e.g., bar charts, line graphs, pie charts).	like brightness, contrast, hue/saturation, and filters and how they impact the appearance of an image. • **Layers**: Pupils know the function and benefits of using layers in image editing software.
Disciplinary Knowledge (The skills and approaches that students will develop)	• **Log in and navigate** school systems confidently and responsibly. • **Use email tools effectively** to manage communication (e.g., folders, CC/BCC, replying appropriately). • **Create and format documents and presentations** to share information clearly and professionally. • **Share and collaborate** on resources using platforms like OneNote. • **Practice safe and ethical behaviour** online, including	• **Classify and differentiate** between hardware and software components. • **Describe and explain** how individual components contribute to the overall function of a computer system. • **Analyse** the purpose and suitability of different input and output devices for various tasks or user needs. • **Interpret and articulate** the role of the CPU and how it processes data and instructions. • **Compare** types of software and	• **Plan programs** effectively by designing flowcharts or writing algorithms before coding. • **Build functional Scratch programs** using block code to meet a specific goal or solve a problem. • **Sequence commands** accurately to ensure correct program flow. • **Create and manipulate variables** to store and change data during program execution. • **Design programs that interact** with the user through input features (e.g., keyboard or mouse input). • **Use selection and logic** to control decision-making within the program. • **Apply Boolean operators** to refine conditions and control flow. • **Test and debug code** independently to ensure functionality and correct any issues.	• **Manipulate and manage data** confidently within a spreadsheet environment. • **Apply formatting** thoughtfully to make data clearer and more meaningful for the audience. • **Construct and apply formulas** to solve problems and automate calculations. • **Use functions effectively** to simplify tasks, analyse data, and derive insights. • **Choose and create appropriate graphs** to communicate data findings clearly.	• **Use selection tools** accurately to isolate parts of an image (e.g., marquee, lasso, magic wand). • **Edit and manipulate images** by removing unwanted elements cleanly and precisely. • **Apply layers effectively** to build complex compositions in an organised and editable way. • **Combine text and images** to convey a clear message or aesthetic, considering layout and design principles. • **Apply adjustments and filters** to enhance or stylise

	protecting passwords and personal information. • **Evaluate online information** for reliability and relevance using search strategies. • **Reflect critically** on how social media impacts communication, privacy, and well-being.	evaluate their uses in real-world contexts. • **Explore and reflect** on how operating systems support the running of a computer and manage its resources.		• **Evaluate spreadsheet designs** for efficiency, readability, and effectiveness in presenting data.	images for different purposes. • **Critically evaluate** design choices based on image quality, clarity, copyright, and visual impact.
Assessment (The methods that teachers will use to assess the progress of all students)	• Summative end of unit assessment task (L6) – Completed online, provides pupils with instant feedback on performance	• Summative end of unit assessment task (L7) – Completed online, provides pupils with instant feedback on performance	• Summative assessment task (L5) – Completed online, provides pupils with instant feedback on performance • End of unit review of programming project – graded against specific criteria (L11)	• Summative assessment task (L6) – Completed online, provides pupils with instant feedback on performance	• Summative assessment task (L6) – Completed online, provides pupils with instant feedback on performance
Reading, Writing and Vocabulary	**Key Vocab** Digital Literacy, E-Safety, Social Media, Username, Password, Search Engine, Email, Outlook, OneNote **Disciplinary Reading -** Read an article on data privacy and summarise the key risks using digital literacy vocabulary. **Disciplinary Writing -** Create a how-to guide for younger students explaining how to use	**Key Vocab** Hardware, Software, Input, Output, Random Access Memory, Read Only Memory, Processor Central Processing Unit, Operating System, Application, Binary **Disciplinary Reading -** Read an article explaining how the CPU, RAM, and storage devices interact. Highlight and define all technical terms (CPU, RAM, input/output, etc.).	**Key Vocab** Instruction, Sequence, Selection, Iteration, Algorithm, Variable, Input, Programming, Testing **Disciplinary Writing -** Write a reflection explaining how you debugged an error in your Scratch code.	**Key Vocab** Spreadsheet, Data, Information, Formula, Function, Graph, Cell, Formatting, Calculation, Operator **Disciplinary Reading -** Read a guide on how spreadsheets are used in business. Identify key vocabulary (e.g., cell, formula, function, operator).	**Key Vocab** Graphics, Editing, Bitmap, Vector, Selection, Properties, Pixel, Pixelated, Resolution, Sourcing, Copyright **Disciplinary Reading -** Read an article comparing bitmap and vector graphics. Create a glossary of terms. **Disciplinary Writing -** Write an evaluation of the graphics editing task

	email and cloud tools safely.			**Disciplinary Writing -** Write a set of instructions for a classmate explaining how to use a function like VLOOKUP or SUM.	you completed, using terms like "layers," "selection tool," "adjustments," etc.
Numeracy	• **Time management** – managing schedules in Outlook or OneNote; interpreting time and dates. • **Interpreting search results** – evaluating numerical data such as search rankings, number of hits, or file sizes.	• **Logical reasoning and pattern recognition** – understanding how the CPU processes instructions and how systems follow sequences. • **Basic understanding of data representation** (optional enrichment) – e.g., binary or memory/storage sizes (bits, bytes).	• **Sequencing and order** – understanding the impact of instruction order on program logic. • **Use of coordinates** – moving sprites in a 2D plane using x and y values. • **Variables and data handling** – storing and manipulating numerical values. • **Boolean logic** – using comparison operators (=, >, <) and logical connectives (and, or, not). • **Debugging with precision** – identifying and correcting numerical errors in logic or timing.	• **Arithmetic operations** – using addition, subtraction, multiplication, and division in formulas. • **Understanding of functions** – applying SUM, AVERAGE, MAX, MIN, COUNT, IF, and VLOOKUP with numerical data. • **Data analysis** – interpreting numerical patterns, comparing values, and evaluating outputs. • **Graphical representation of data** – choosing and interpreting bar charts, line graphs, and pie charts.	• **Measurement and proportion** – understanding image resolution (e.g., pixels per inch), resizing images while maintaining aspect ratio. • **Scale and dimension** – adjusting image size while keeping quality consistent.
Personal Development	• **Online safety assembly** – Being a good digital citizen.	• N/A	• **Coding Club** – Available for pupils to embed and enhance their programming skills at lunch time.	• N/A	• **Poster Competition** – Provides the opportunity for pupils to try out their graphics editing skills to create a poster for a specific theme

SUBJECT Curriculum Map

Year 8

Rationale and Links to The National Curriculum
Computing skills are a vital component to any student's future success when they leave Carnforth High School. Through engaging content and a supportive environment, pupils will develop the skills and understanding that allow them to safely and confidently use technology in the real world; building their digital literacy, helping them become good "digital citizens" and preparing them for further study or job opportunities in the technology sector and beyond. At the heart of Computing is Computer Science, where students learn the key ideas behind information and computation, how digital systems function, and how to apply this knowledge through programming. With this foundation, they're able to confidently use information technology to design programs, build systems, and create all kinds of digital content. The design of the Key Stage 3 Computing curriculum revolves around ensuring pupils are equipped with the relevant skills that will enable them to successfully study either GCSE Computer Science or Creative iMedia as an option for Key Stage 4. It takes inspiration from a wide range of sources within the education sector, building on the experience of a number of professional bodies, such as the NCCE, PGOnline, CSUK and more, while focussing on providing content that is relatable to our pupils. The content in Year 8 allows pupils to build on their experience in Year 7 in meeting the skills and processes outlined in the Computing National Curriculum. Unit 1 revolves around understanding a range of ways to use technology safely, respectfully, responsibly and securely, Unit 2 looks at understanding algorithms and computational thinking, Unit 3 and 4 give the opportunity to design, use and evaluate computational abstractions and introduces pupils to their second programming language (Python), Unit 5 and 6 allow pupils to undertake creative projects that involve selecting, using, and combining multiple applications, collecting and analysing data, create, re-use, revise and re-purpose digital artefacts.

	Half Term 1 – Cyber Security and Digital Wellbeing (6 Lessons)	**Half Term 2 – Computational Thinking (7 Lessons)**	**Half Term 3 & Half Term 4 – Python Programming (11 Lessons)**	**Half Term 5 – Databases (6 Lessons)**	**Half Term 6 – Audio Editing (6 Lessons)**
Key Topics	• Health & safety issues when using technology • Impact of online abuse and false information	• Using algorithms to help solve problems • Understand how to utilise abstraction • Understand how to utilise decomposition	• Use a text-based programming language to create programs • Effective planning, including the use of flowchart and pseudocode algorithms • Using different data types • Create and assign variables	• Understand the difference between flat file and relational databases • Create tables in a database	• Understand the advantages of different audio file formats (MP3 and WAV) • Creating a script for audio projects

	• Understand different types of online scams • Understand different types of computer misuse • Methods used to stop network threats • Careers in cyber security	• Understand how to utilise pattern recognition • Demonstrating problem solving skills • Converting binary, denary and hexadecimal values	• Identify and fix syntax and logic errors • Use sequence, selection and iteration to create working programmes • Utilise comments to document code in a programme • Create a programme to meet the needs of a given problem	• Assign appropriate properties to tables and • Be able to add, edit and delete data in a database table • Be able to query data in tables • Utilise basic SQL commands to retrieve data from a database	• Sourcing assets appropriately • Using basic editing tools, such as; importing, selecting, cutting, zooming, timeshift • Using advanced editing tools, such as; fading-in/out, changing pitch and tempo, envelope, amplification • Collecting audio using a microphone
Substantive Knowledge (The knowledge the students will develop)	• **Health & safety when using technology:** Pupils know risks like eye strain, posture issues, and repetitive strain injury, and how to prevent them. • **Online abuse and misinformation:** Knowing the forms of online abuse (e.g., cyberbullying, trolling), and how false information spreads online (e.g., fake news, deepfakes). • **Online scams:** Recognising different scams such as phishing, identity theft, and fake websites. • **Computer misuse:** Pupils know what constitutes illegal or	• **Algorithms:** Pupils know what an algorithm is and how it provides a step-by-step method for solving **problems.** • **Abstraction:** Pupils know how to filter out unnecessary information to focus on what's important. • **Decomposition:** Knowing how to break down complex problems into smaller, manageable parts. • **Pattern recognition:** Pupils know how to identify trends, similarities, or repeated elements in problems or data. • **Number systems:** Pupils know binary, denary (decimal), and hexadecimal systems, and how to	• **Text-based programming:** Pupils know how Python differs from block-based languages, focusing on syntax, indentation, and structure. • **Algorithms:** Pupils know how to plan with flowcharts and pseudocode before coding. • **Data types:** Knowing and using different data types (e.g., string, integer, float, boolean). • **Variables:** Pupils know how to declare and assign variables to store data. • **Control structures:** Knowing how to use: **a) Sequence** – step-by-step instructions b) **Selection** – decisions using if/else **c) Iteration –** loops using for and while • **Error types:** Pupils know the difference between syntax errors (e.g., typos) and logic errors (e.g., wrong output). • **Code documentation:** Pupils know the purpose and use of comments (#) to explain or annotate code. • **Problem-solving approach:** Pupils know how to create programs that meet a brief or specification.	• **Flat file vs relational databases:** Pupils know the structural differences, benefits, and drawbacks of each model. • **Tables and fields:** Knowing how databases are organised into tables with fields (columns) and records (rows). • **Data types and properties:** Pupils know how to assign appropriate field types (e.g., text, number, date/time) and properties (e.g., primary key, validation rules). • **SQL basics:** Pupils know how SQL (Structured Query Language) is used to retrieve, filter, and organise data.	• **Audio file formats:** Pupils know the differences between formats like MP3 and WAV, including compression, quality, and file size. • **Scripting for audio:** Knowing the importance of planning dialogue, narration, or structure before recording. • **Sourcing assets:** Pupils know how to find and use sound files legally and ethically, including copyright considerations. • **Basic audio editing tools:** a) Importing b) Selecting c) Cutting d) Zooming e) timeshift (moving clips) • **Advanced audio editing tools:**

	unethical computer use, including hacking, malware distribution, and unauthorised access. • **Cybersecurity methods:** Knowing common ways to prevent network threats, such as firewalls, antivirus software, strong passwords, and multi-factor authentication. • **Cyber Careers:** Pupils know the range of careers available in cyber and the requirements needed to persue these careers	convert between them.		• **Queries:** Knowing how to extract specific information from one or more tables using filters and conditions.	a) Fading in/out b) pitch shifting c) tempo adjustments d) envelope control e) amplification • **Microphone use:** Knowing how to record clean audio, including mic technique and environment awareness.
Disciplinary Knowledge (The skills and approaches that students will develop)	• **Identify and evaluate risks** associated with technology use on physical, emotional, and societal levels. • **Critically assess online content** for credibility, reliability, and bias. • **Recognise and respond to scams** and security threats in realistic digital scenarios. • **Analyse legal and ethical issues** surrounding misuse of technology.	• **Design and write algorithms** to solve specific tasks or challenges. • **Apply abstraction** to simplify real-world problems by ignoring irrelevant details. • **Decompose problems** logically to make them easier to solve. • **Use pattern recognition** to speed up problem-solving or spot reusable solutions. • **Demonstrate logical problem-solving** in various contexts (not	• **Write and test Python code** using correct syntax and structure. • **Plan programs effectively** using pseudocode and flowcharts. • **Use and manipulate variables** with appropriate data types. • **Apply control structures** (sequence, selection, iteration) to build functional programs. • **Debug code** by identifying and fixing syntax or logic errors. • **Document code** clearly using comments for readability and maintenance. • **Solve real problems** by writing programs that meet specific user needs or briefs.	• **Design and create** tables with appropriate field names, data types, and primary keys. • **Input and maintain data** by adding, editing, and deleting records in tables. • **Assign field properties** correctly to improve data integrity and prevent errors. • **Construct queries** to search for and filter data using multiple criteria.	• **Compare and evaluate** file formats based on project needs (e.g., quality vs file size). • **Write effective scripts** for podcasts, voiceovers, or audio stories. • **Source and organise** audio assets legally, with attention to quality and compatibility. • **Use editing software** (e.g., Audacity) to: a) Import and arrange clips b) Cut and refine recordings

	• **Evaluate security measures** for effectiveness in protecting personal and network data. • **Be able to search and navigate** job listings to find suitable job roles of interest	limited to programming). • **Convert between number systems** fluently and understand their relevance in computing (e.g., colour codes, IP addresses, memory sizes).		• **Write and execute basic SQL commands, such as:** a) SELECT to retrieve fields b) WHERE to filter records c) ORDER BY to sort results	c) Shift clips to align with a timeline • **Apply advanced editing techniques** to enhance audio quality and creativity. • **Record audio** using a microphone effectively, ensuring clarity and minimal background noise.
Assessment (The methods that teachers will use to assess the progress of all students)	• Summative end of unit assessment task (L6) – Completed online, provides pupils with instant feedback on performance	• Summative end of unit assessment task (L7) – Completed online, provides pupils with instant feedback on performance	• Summative assessment task (L5) – Completed online, provides pupils with instant feedback on performance • End of unit review of programming project – graded against specific criteria (L11)	• Summative assessment task (L6) – Completed online, provides pupils with instant feedback on performance	• Summative assessment task (L6) – Completed online, provides pupils with instant feedback on performance
Reading, Writing and Vocabulary	**Key Vocab** Repetitive Strain injury, Cyber Bullying, Misinformation, Fake News, Data Protection, Password, Copyright, Environment, Cyber, Security, Phishing, Ransomware, Malware, Virus, Hacking, Social Engineering, Authentication, Anti-Virus, Firewall, Biometrics, Computer Misuse Act **Disciplinary Reading -** Read an article on the impact of cybercrime.	**Key Vocab** Algorithm, Flowchart, Pseudocode, Abstraction, Decomposition, Pattern Recognition, Binary, Denary, Hexadecimal **Disciplinary Reading -** Read a section from a textbook on abstraction, decomposition, and pattern recognition. **Disciplinary Writing -** Write a summary comparing how abstraction and decomposition were used to solve a class problem.	**Key Vocab** Algorithm, Flowchart, Pseudocode, Python, Integrated Development Environment (IDE), Programming, Variable, Selection, Iteration, Decision, Sequencing, Syntax, Logic Error, Debug, Comments **Disciplinary Reading -** Read a code walkthrough and highlight all keywords (e.g., variable, string, loop, function). **Disciplinary Writing -** Write a code explanation paragraph, clearly describing how your program works and why you chose certain structures.	**Key Vocab** Flat-file, Relational, Database, Table, Field, Record, Entity, Query, Structured Query Language (SQL), Boolean, Primary Key , Foreign Key **Disciplinary Reading -** Read a database case study and extract terminology such as "table," "field," "primary key," and "query". **Disciplinary Writing -** Write a user guide explaining how to perform a basic query in SQL.	**Key Vocab** Audio, Audacity, Import, Record, Select, Zoom, Cut, Fade, Amplify, Timeline, Pitch, Tempo, Copyright, MP3, WAV **Disciplinary Reading -** Read a beginner's guide to audio editing tools and highlight vocabulary like "amplify," "fade," "clip," "pitch," and "tempo." **Disciplinary Writing -** Write a blog post describing the process you followed to edit an audio file.

	Disciplinary Writing - Write a guide to staying safe when using computers.				
Numeracy	• **Data interpretation** – e.g., reading statistics about cyberbullying incidents, scam reports, or environmental impact (e.g., % of e-waste recycled). • **Evaluating energy usage** – interpreting numerical data on electricity consumption or carbon emissions linked to device use.	• **Working with number systems –** binary (base 2), decimal (base 10), hexadecimal (base 16). • **Conversions and calculations –** converting between binary, decimal, and hex using division, multiplication, and place value understanding. • **Logic and reasoning** – using structured, sequential thinking to follow or create algorithms.	• **Working with numeric data types –** integers and floats for arithmetic and logic-based tasks. • **Arithmetic and expressions –** using operators (+, -, *, /, %) within programs. • **Logical reasoning** – applying Boolean logic (and, or, not) and relational operators (<, >, ==). • **Using iteration effectively –** understanding loops as controlled repetition, often tied to numeric counters or conditions.	• **Logical operators –** using =, >, <, >=, AND, OR in queries and SQL commands. • **Sorting and filtering data –** interpreting structured data sets and applying logical rules to extract relevant information. • **Understanding data types** – choosing between text, number, date, etc., and recognising how they behave in operations.	• **Timing and duration management –** accurately trimming clips and syncing them to a time-based structure (e.g., seconds, minutes). • **Volume and amplification levels –** adjusting decibel levels logically to balance tracks. • **Tempo and pitch changes –** understanding numerical values (e.g., bpm, semitone shifts) when modifying audio.
Personal Development	• **Online safety assembly** – Being a good digital citizen.	• N/A	• **Coding Club** – Available for pupils to embed and enhance their programming skills at lunch time.	• N/A	• N/A

SUBJECT Curriculum Map

Year 9

Carnforth High School

Rationale and Links to The National Curriculum
Computing skills are a vital component to any student's future success when they leave Carnforth High School. Through engaging content and a supportive environment, pupils will develop the skills and understanding that allow them to safely and confidently use technology in the real world; building their digital literacy, helping them become good "digital citizens" and preparing them for further study or job opportunities in the technology sector and beyond. At the heart of Computing is Computer Science, where students learn the key ideas behind information and computation, how digital systems function, and how to apply this knowledge through programming. With this foundation, they're able to confidently use information technology to design programs, build systems, and create all kinds of digital content. The design of the Key Stage 3 Computing curriculum revolves around ensuring pupils are equipped with the relevant skills that will enable them to successfully study either GCSE Computer Science or Creative iMedia as an option for Key Stage 4. It takes inspiration from a wide range of sources within the education sector, building on the experience of a number of professional bodies, such as the NCCE, PGOnline, CSUK and more, while focussing on providing content that is relatable to our pupils. The content in Year 9 allows pupils to build on their experience from Year 7 and Year 8 in meeting the skills and processes outlined in the Computing National Curriculum. Unit 1 looks at how technology is used to solve real-world problems, Unit 2 looks at how data of various types can be represented and manipulated digitally, Unit 3 and 4 give the opportunity to design, use and evaluate computational abstractions and introduces pupils to alternative programming languages (HTML, CSS and JavaScript), Unit 5 looks at how computers communicate with one another and with other systems, and Unit 6 allows pupils to undertake creative projects that involve selecting, using, and combining multiple applications, collecting and analysing data, create, re-use, revise and re-purpose digital artefacts.

	Half Term 1 – Impact of Digital Technology (6 Lessons)	Half Term 2 – Data Representation (7 Lessons)	Half Term 3 & Half Term 4 – Web Design (11 Lessons)	Half Term 5 – Computer Networks (6 Lessons)	Half Term 6 – Mobile App Development (6 Lessons)
Key Topics	• Uses of artificial intelligence • Technology utilised for self-driving cars • Development of robotics	• Recap converting binary, denary and hexadecimal values • Adding 2 binary values and the issue of overflow • Using Boolean logic	• Understand what makes a good website • Create a web page • Using HTML coding to add basic content to webpages • Create working hyperlinks • Utilise suitable navigation for a website • Use CSS coding to consistently format style options	• What is a network? • Advantages and disadvantages of using computer networks • How data is sent along a network	• Design a health tracker mobile app • Understand mobile phone hardware • Online safety in app development • Develop a mobile app

	• Application of technology in medicine • VR vs AR • Environmental impact of tech	• Converting binary to text using character sets • Converting binary to images • Converting binary to sound • Use of lossy and lossless compression	• Use JavaScript to add interactive elements to a webpage	• Using LANs • Using WANs • Identify different network topologies • Network hardware	• Evaluate a mobile app
Substantive Knowledge (The knowledge the students will develop)	• **Artificial Intelligence (AI)**: Pupils know what AI is, its current capabilities, and common uses (e.g., chatbots, image recognition, predictive text). • **Self-driving car technology**: Knowing the components involved (sensors, AI, machine learning, GPS, LiDAR) and how they work together. • **Robotics development**: Pupils know how robots are built, programmed, and used in sectors like manufacturing, delivery, and exploration. • **Medical technology**: Exploring how digital tools (e.g., diagnostic AI, robotic surgery, wearable tech) improve patient care and healthcare systems.	• **Number systems recap**: Pupils know and converting between **binary**, **denary (decimal)**, and **hexadecimal**. • **Binary addition**: Performing binary addition and Pupils know the concept of **overflow** when a result exceeds the available bit limit. • **Boolean logic:** Pupils know how AND, OR and NOT gates are used in circuits. • **Character sets**: Pupils know how **binary represents text** using character encoding systems like **ASCII** and **Unicode**. • **Binary representation of images**: Knowing	• **Effective web design principles**: Pupils know what makes a website **user-friendly**, **accessible**, **visually appealing**, and **functional** (e.g., layout, colour schemes, typography, responsiveness). • **HTML (HyperText Markup Language)**: Learning how HTML structures content using elements like `<head>`, `<body>`, `<h1>`, `<p>`, `<img>`, and `<a>`. • **Hyperlinks and navigation**: Pupils know how to link pages using anchor tags (`<a href="">`) and how navigation impacts user experience. • **CSS (Cascading Style Sheets)**: Knowing how CSS styles content, including rules for **colour**, **fonts**, **layout**, and **consistency across pages**. • **JavaScript basics**: Pupils know how JavaScript can make websites **dynamic** and **interactive** (e.g., buttons, image sliders, popups).	• **What a network is**: Pupils know that a computer network is two or more devices connected to share data, resources, and communication. • **Advantages & disadvantages**: a) Benefits: file sharing, communication, resource sharing (e.g. printers), central backups. b) Risks: security issues, technical faults, network congestion. • **Data transmission**: Know how data is broken into packets, sent over cables/wirelessly, and reassembled at its destination using protocols. • **LAN (Local Area Network)**: Pupils know networks	• Pupils know how to **design a mobile app**, including planning features for a health tracker. • Pupils know the **main components of mobile phone hardware** (e.g., sensors, touchscreen, memory, processors) and how they relate to app functionality. • Pupils know the importance of **online safety in app development**, including data protection and secure handling of personal information. • Pupils know the process of **developing a mobile app** using design and coding tools. • Pupils know how to **evaluate an app**

	• **Environmental impact**: Pupils know how digital tech contributes to environmental issues (e-waste, energy use) and its potential to help (e.g., smart grids, climate modelling). • **Careers in tech**: Awareness of job roles in the digital sector (e.g., software engineer, data analyst, cybersecurity specialist, AI researcher).	how images are stored using binary (e.g., **bitmaps**, pixels, colour depth). • **Binary representation of sound**: Pupils know how sound is digitised using **sampling rate**, **bit depth**, and binary encoding. • **Compression techniques**: Pupils know the difference between **lossy** (e.g., MP3, JPEG) and **lossless** (e.g., PNG, FLAC) compression, including advantages and drawbacks.		within a limited area (e.g. schools, offices). • **WAN (Wide Area Network)**: Pupils know larger-scale networks like the internet that span multiple locations. • **Network topologies**: Star, Bus, Ring, Mesh – advantages, disadvantages, and how data flows through each. • **Network hardware**: Pupils know the roles of devices such as: Routers, switches, hubs, modems, network interface cards (NICs), access points.	against criteria such as usability, accessibility, performance, and safety.
Disciplinary Knowledge (The skills and approaches that students will develop)	• **Evaluate the benefits and risks** of emerging technologies in everyday life. • **Research and interpret information** about how different technologies are used in specific industries. • **Reflect critically** on the ethical, social, and environmental implications of	• **Convert between number systems** with confidence (binary ↔ denary ↔ hex). • **Perform binary addition** and identify **overflow errors** in calculations. • **Use Boolean logic,** identifying AND, OR and NOT gates and use these in a circuit diagram.	• **Design web pages** using effective layout and navigation structures. • **Write and edit HTML code** to add text, images, headings, and links to a webpage. • **Create working hyperlinks** that connect pages or link to external sites. • **Build a navigation menu** that is clear, consistent, and easy to use. • **Apply CSS styling** to customise fonts, colours, spacing, and layout across multiple pages. • **Incorporate JavaScript** to add interactivity (e.g., form validation, button actions, hover effects). • **Test and debug** their website to ensure it functions as intended and displays correctly across pages.	• **Compare different network types** and explain suitable use cases (e.g. LAN vs WAN). • **Analyse advantages and disadvantages** of network setups and topologies. • **Identify appropriate network hardware** based on the needs of a scenario. • **Draw and interpret network diagrams**,	• **Plan and design** a mobile app using flowcharts, storyboards, or mock-ups. • **Link app features** to the hardware capabilities of mobile devices (e.g., accelerometer for step counting). • **Apply principles of online safety** and responsible design to protect users' data. • **Build and test** a working mobile app,

	technological advancement. • **Explore career paths** and develop awareness of required skills and qualifications in the tech sector. • **Analyse case studies** to understand real-world applications of AI, robotics, and medical technology.	• **Interpret character codes** to convert binary values into readable text. • **Explain how images and sound** are digitally represented and stored using binary. • **Compare compression methods** and justify when to use lossy vs lossless formats. • **Apply logical reasoning** to understand how digital information is manipulated at a low level.		showing how devices are connected. • **Describe the flow of data** through a network using correct terminology (e.g., packets, protocol, IP address). • **Evaluate network performance factors** such as bandwidth, topology, and hardware.	iterating and refining design. • **Evaluate digital products**, providing constructive, evidence-based feedback using subject-specific vocabulary.
Assessment (The methods that teachers will use to assess the progress of all students)	• Summative end of unit assessment task (L6) – Completed online, provides pupils with instant feedback on performance	• Summative end of unit assessment task (L7) – Completed online, provides pupils with instant feedback on performance	• Summative assessment task (L5) – Completed online, provides pupils with instant feedback on performance • End of unit review of programming project – graded against specific criteria (L11)	• Summative assessment task (L6) – Completed online, provides pupils with instant feedback on performance	• Summative assessment task (L6) – Completed online, provides pupils with instant feedback on performance
Reading, Writing and Vocabulary	**Key Vocab** Artificial Intelligence, Autonomous, Sensors, LiDAR, Turing-Test, Robotics, Medicine, Virtual Reality, Augmented Reality, Environmental, Application	**Key Vocab** Binary, Denary, Boolean, Overflow, Storage, Character Set, ASCII, Unicode, Bitmap, Pixel, Pixelation, Analogue, Sample, Amplitude, Frequency, Compression, Lossy, Lossless	**Key Vocab** Website, Webpage, Structure, Static, Hyperlink, Navigation, HTML, CSS, Style, Formatting, Tags, JavaScript **Disciplinary Reading -** Read a tutorial on HTML/CSS/JavaScript and identify new terms (e.g., tag, hyperlink, class, style).	**Key Vocab** Network, Data, Packet, Local Area Network, Wide Area Network, Topology, Star, Bus, Ring, Router, Ethernet, Wi-Fi, Hub, Switch **Disciplinary Reading -** Read a description of	**Key Vocab** Mobile application, User interface, User experience, Prototype, Wireframe, Storyboard, Sensor, Touchscreen, Hardware, Online safety, Data protection, Evaluation, Accessibility, Feature set, Deployment

	Disciplinary Reading - Read an article about AI or robotics in society and list pros and cons using subject-specific vocabulary. **Disciplinary Writing -** Write a summary on whether technology is having a positive of negative impact on the world.	**Disciplinary Reading -** Read a chapter or guide on binary and hexadecimal. Create a vocabulary map linking terms like "overflow," "character set," "lossless," etc. **Disciplinary Writing -** Write a paragraph explaining how an image is represented in binary.	**Disciplinary Writing -** Write a developer log entry describing the structure and features of your website project.	LAN, WAN, and topologies. Create a vocabulary list with definitions. **Disciplinary Writing -** Write an explanation of how data travels across a network using correct terminology.	**Disciplinary Reading -** Read a case study about health or fitness apps. Extract vocabulary such as "interface," "hardware," "evaluation," and "safety." **Disciplinary Writing -** Write an evaluation report of your health tracker app, using disciplinary vocabulary to explain design choices and user safety features.
Numeracy	• **Analysing environmental data** – interpreting graphs or numerical information about e-waste, energy consumption, or CO_2 emissions. • **Technology performance metrics** – comparing specifications or success rates (e.g., accuracy of AI, error rates in autonomous driving).	• **Place value systems** – applying positional notation in binary, denary, and hexadecimal. • **Binary addition** – performing calculations and understanding carry-over logic. • **Understanding bit depth and resolution** – calculating file sizes for images or sound based on bits per pixel/sample.	• **Using coordinates and units** – applying percentages, pixels, `em`, and `rem` for layout and spacing in CSS. • **Understanding RGB and HEX colour codes** – interpreting and manipulating numeric colour representations.	• **Calculating data transfer speed and bandwidth** – comparing network speeds (e.g. Mbps, Gbps). • **Working with topologies** – logical thinking about how connections scale (e.g. mesh topology connections: $n(n-1)/2$). • **Binary in addressing** – understanding basics of IP addressing and subnetting (introductory level).	• **Using numerical data** within a health tracker app (e.g., counting steps, measuring distance, tracking calories). • **Applying formulas** for calculations within the app • **Logical reasoning** in app design, including conditional logic (if/then decisions).
Personal Development	• **Online safety assembly** – Being a good digital citizen.	• N/A	• **Coding Club** – Available for pupils to embed and enhance their programming skills at lunch time.	• N/A	• N/A

SUBJECT Curriculum Map

Year 10 – GCSE Computer Science

Carnforth High School

Rationale and Links to The National Curriculum
Studying GCSE Computer Science provides students with the knowledge and skills needed to understand and shape the digital world around them. Technology is at the core of modern life, from smartphones and apps to healthcare systems, transportation, and entertainment. By learning how computers work and how software is created, students gain insight into the foundations of innovation and problem-solving in the 21st century. The course develops both practical and analytical skills. Programming encourages logical thinking, creativity, and resilience, as students design, test, and improve solutions to real problems. Understanding computer systems, data, and networks also helps students become informed users and creators of technology, not just passive consumers. These skills are highly transferable, supporting success in mathematics, science, and many other disciplines. GCSE Computer Science is also an excellent preparation for future opportunities. With the growing demand for computing expertise in almost every industry, the subject opens pathways to further study and a wide range of careers, from software development and cybersecurity to engineering and data science. Most importantly, it empowers students to engage confidently with the digital world and to contribute positively to the technological future. Pupils are given the opportunity to study GCSE Computer Science in Year 10 and Year 11, which helps them to develop their capability, creativity and knowledge in computer science, develop and apply their analytic, problem-solving, design, and computational thinking skills, understand how changes in technology affect safety, including new ways to protect their online privacy and identity, and how to identify and report a range of concerns.

	Half Term 1 - Computer Systems (15 Lessons)	**Half Term 2 – Programming Fundamentals (17 Lessons)**	**Half Term 3 – Computational Thinking (Algorithms) (14 Lessons)**	**Half Term 4 – Programming Next Steps (12 Lessons)**	**Half Term 5 – Computational Thinking (Data Representation) (14 Lessons)**	**Half Term 6 – Computer Networks & Cyber Security (14 Lessons)**
Key Topics	• What is a computer, including embedded systems • Understanding the CPU role • What affects the performance of the CPU	• Using data types • Assigning variables • Assigning constants • Writing programs using selection • Creating input and output	• Understand the term algorithm • Understand the use of decomposition & abstraction • Representing algorithms using pseudocode and flowcharts	• Creating programs using iteration (While and For loops) • Using trace tables to follow a loop • Nested selection and iteration • Methods of data validation	• Use the different number bases; decimal, binary and hex • Understand how binary and hex are used • Convert values between the	• Advantages & disadvantages of networks • PAN, LAN and WAN • Wired vs wireless • Topologies – star vs bus • Protocols

	• Primary memory, including; RAM, ROM, Cache and Registers • Secondary storage, including; Optical, magnetic, solid state and Cloud • Application vs system software • The roles of the operating system • Identify and utilise logic gate symbols to create logic diagrams	• Generating random numbers • Using arithmetic operators • Using logical expressions • Use trace tables to predict the outputs of a program	• Be able to read and follow algorithms, in terms of input, process and output • Understand the purpose of different algorithms • Use and compare linear and binary search • Use and compare the merge and bubble sort	• Using pseudocode to plan a program • Creating programs using subroutines • Using functions • Using arrays • Using records	different number bases • Understand the different levels of size; bit, byte, KB, MB, GB, TB • Perform binary arithmetic • Apply binary shifts • Understand the use of character sets • Representing images • Representing sound • Data compression	• Understand the uses of common protocols • Describe the 4 layer TCP/IP model • Importance of cyber security • Understand the key threats to a network • Understand use of penetration testing • Understand the use of social engineering • Understand the key security measures
Substantive Knowledge (The knowledge the students will develop)	• **What a computer is**, including the distinction between general-purpose and **embedded systems** (e.g., washing machines, satnavs). • **The role of the CPU**: Pupils know that the CPU processes instructions using the fetch-decode-execute cycle. • **Factors affecting CPU performance**: Pupils know how clock speed, number of cores, and cache size impact processing power. • **Primary memory**: Pupils know the purpose and differences between	• **Data types**: Pupils know the common types used in programs, such as **integer**, **float**, **string**, **Boolean**, and **character**. • **Variables**: Pupils know how to declare and assign **variables** to store values during program execution. • **Constants**: Pupils know how to assign **constant values** that do not change during program execution. • **Selection**: Pupils know how to write programs that use **if**, **elif**, and **else** statements to make decisions.	• **Algorithms**: Pupils know that an algorithm is a **step-by-step procedure** for solving a problem or performing a task. • **Decomposition**: Pupils know how to break complex problems into **smaller, more manageable parts**. • **Abstraction**: Pupils know how to **filter out unnecessary details** to focus on what's important in solving a problem. • **Pseudocode and flowcharts**: Pupils know how algorithms can be **represented visually or in structured text**	• **Iteration**: Pupils know how to use **WHILE** and **FOR loops** to repeat sections of code based on conditions or ranges. • **Trace tables and loops**: Pupils know how to use **trace tables to track values** through iterative loops and check for correct output. • **Nested control structures**: Pupils know how to use **selection and iteration inside each other**, such as IF statements inside loops or loops within loops.	• **Number bases**: Pupils know the purpose of **decimal (denary), binary, and hexadecimal** in computing contexts. • **Binary and hexadecimal usage**: Pupils know how binary is used in **data storage and processing**, and how hexadecimal is used for **memory addresses and compact representation**. • **Conversions**: Pupils know how to **convert values** between **binary**, **hex**, and **decimal**. • **Units of data**: Pupils know the relationship between	• **Network types**: Pupils know the differences between **PAN (Personal Area Network), LAN (Local Area Network)**, and **WAN (Wide Area Network)**. • **Benefits and drawbacks**: Pupils know the **advantages and disadvantages** of using computer networks. • **Wired vs wireless**: Pupils know the **characteristics, benefits, and limitations** of wired and wireless network connections. • **Network topologies**: Pupils know how **star**

	RAM, **ROM**, **Cache**, and **Registers** in the memory hierarchy. • **Secondary storage**: Pupils know the characteristics, advantages and disadvantages of **optical**, **magnetic**, **solid-state**, and **cloud** storage solutions. • **Application vs system software**: Pupils know the difference between user-facing programs (e.g., Word, browsers) and software that supports hardware operation (e.g., device drivers, file management). • **The role of the operating system**: Pupils know how the OS manages memory, multitasking, input/output devices, and user interfaces. • **Logic gates**: Pupils know the symbols and functions of **AND**, **OR**, and **NOT** gates, and how to draw simple logic diagrams using them.	• **Input and output**: Pupils know how to use programming syntax to **receive input from a user** and **display output** to the screen. • **Random number generation**: Pupils know how to use built-in functions to generate **random numbers**, typically for games or simulations. • **Arithmetic operators**: Pupils know how to use **+**, **-**, *****, **/**, **%** (modulus), and ****** (exponentiation) within programs. • **Logical expressions**: Pupils know how to use Boolean logic (e.g., **and**, **or**, **not**) to form expressions that control program flow. • **Trace tables**: Pupils know how to **use trace tables** to track variable changes and **predict output** of a program step-by-step.	to aid understanding and planning. • **Input–Process–Output**: Pupils know how to **identify and interpret the flow** of data through an algorithm. • **Types of algorithms**: Pupils know the purpose and application of algorithms for **searching and sorting** data. • **Search algorithms**: Pupils know how **linear search** checks items one by one, and how **binary search** efficiently splits a sorted list in half. • **Sort algorithms**: Pupils know how **bubble sort** repeatedly compares and swaps adjacent values, while **merge sort** splits and recombines data more efficiently.	• **Data validation**: Pupils know methods such as **length checks, range checks, type checks, and presence checks** to ensure correct input. • **Pseudocode planning**: Pupils know how to **use pseudocode to design** the logic of a program before coding. • **Subroutines**: Pupils know how to **structure programs using subroutines** to organise code and avoid repetition. • **Functions**: Pupils know how to create and use **functions that return values**, including how to pass parameters. • **Arrays**: Pupils know how to use **arrays to store and access multiple values** using indexed positions. • **Records**: Pupils know that **records** store related data items of different types under one structure (like database entries or object-like storage).	**bit, byte, kilobyte (KB), megabyte (MB), gigabyte (GB), and terabyte (TB)**. • **Binary arithmetic**: Pupils know how to **add binary numbers** and recognise the issue of **overflow**. • **Binary shifts**: Pupils know how to perform **left and right shifts** and understand their effects (e.g., multiplying or dividing by 2). • **Character sets**: Pupils know how characters are represented using standards like **ASCII** and **Unicode**. • **Image representation**: Pupils know how images are stored using **pixels, resolution, colour depth**, and **file size**. • **Sound representation**: Pupils know how sound is represented through **sampling rate, bit depth, and sample resolution**. • **Data compression**: Pupils know the purpose of compression and the	and **bus** topologies are structured and the **advantages/disadvantages** of each. • **Protocols**: Pupils know what **network protocols** are and the purposes of common examples such as **HTTP, HTTPS, FTP, TCP/IP, SMTP, IMAP**. • **TCP/IP model**: Pupils know the structure and function of the **4-layer TCP/IP model** (Application, Transport, Internet, Link). • **Cyber security**: Pupils know the importance of protecting systems and data from cyber threats. • **Network threats**: Pupils know the nature of common threats including **malware, phishing, brute force attacks, denial of service, data interception**, etc. • **Penetration testing**: Pupils know that penetration testing is used to **identify and fix vulnerabilities** in systems.

					difference between **lossy and lossless methods**.	• **Social engineering**: Pupils know how manipulation techniques (e.g. phishing, pretexting) are used to gain access to systems or data. • **Security measures**: Pupils know key defences such as **firewalls, user access levels, encryption, anti-malware software**, and **password policies**.
Disciplinary Knowledge (The skills and approaches that students will develop)	• **Label and describe components** of a computer system and explain how they interact. • **Compare types of memory and storage**, justifying appropriate choices for different scenarios. • **Analyse the impact of CPU specifications** on system performance. • **Differentiate between types of software** and explain their roles in a system. • **Construct and interpret logic gate diagrams** to model simple decision-making systems.	• **Declare and use variables and constants** effectively in code. • **Write programs using selection** to control flow based on conditions. • **Read and respond to user input** and display output clearly. • **Incorporate random values** into programs for unpredictability. • **Use arithmetic and logic operators** to perform and evaluate expressions. • **Trace the flow of a program**, predicting and explaining its output with trace tables.	• **Break down problems** using decomposition to manage complexity. • **Remove unnecessary detail** through abstraction to simplify solutions. • **Represent algorithms** using structured pseudocode and flowcharts. • **Interpret and trace algorithms**, identifying inputs, processes, and expected outputs. • **Apply search algorithms** to locate data in lists and justify when to use each. • **Apply and compare sorting algorithms**,	• **Design and implement loops** (FOR/WHILE) for repeated code execution. • **Trace loop execution** using trace tables to predict or debug results. • **Write programs with nested structures**, combining loops and conditional logic. • **Apply appropriate validation techniques** when accepting user input. • **Plan programs** using pseudocode to define logic before implementation. • **Break programs into subroutines**, improving	• **Convert between number bases** accurately and efficiently. • **Perform binary arithmetic**, including addition with carry and identifying overflow. • **Apply binary shifts** to values and describe their effects. • **Interpret character encoding**, including encoding and decoding text using ASCII or Unicode. • **Analyse how changes to resolution or sample rate** affect image/sound quality and file size.	• **Classify networks** based on size and purpose (PAN, LAN, WAN). • **Compare network setups** (e.g. star vs bus) and evaluate the most effective structure for a given scenario. • **Explain the role of protocols** and relate them to specific tasks within networking. • **Break down the TCP/IP model**, linking each layer to real-world applications. • **Identify cyber threats** and assess their impact on individuals or organisations.

	• **Evaluate storage devices** based on capacity, durability, speed, and cost.	• **Debug simple programs** by identifying logic or syntax errors through reasoning.	discussing efficiency and method. • **Evaluate algorithm suitability** for specific problems (e.g. size of data set, performance needs).	modularity and readability. • **Define and use functions**, with and without return values, including parameter use. • **Store and retrieve data using arrays** and **records** to manage collections of information. • **Develop full working programs** that use structured and reusable components.	• **Compare compression methods** and justify the use of lossy vs lossless in different scenarios. • **Break down the components of digital representations**, including calculating image or sound file sizes based on attributes.	• **Evaluate security measures** for different types of cyber attacks. • **Explain and apply cyber security principles**, such as confidentiality and integrity. • **Recognise and assess social engineering techniques** and how to prevent them. • **Interpret scenarios** and suggest appropriate security countermeasures.
Assessment (The methods that teachers will use to assess the progress of all students)	• Mid-unit assessment paper (past exam questions) • End-of-unit assessment paper (past exam questions)	• Mid-unit assessment paper (past exam questions) • End-of-unit assessment paper (past exam questions)	• Mid-unit assessment paper (past exam questions) • End-of-unit assessment paper (past exam questions)	• Mid-unit assessment paper (past exam questions) • End-of-unit assessment paper (past exam questions)	• Year 10 Mock Exam (Paper 1 and Paper 2 combined) • End-of-unit assessment paper (past exam questions)	• Mid-unit assessment paper (past exam questions) • End-of-unit assessment paper (past exam questions)

Reading, Writing and Vocabulary	**Key Vocab** Computer, Embedded system, CPU, Control unit (CU), Arithmetic logic unit (ALU), Clock speed, Cache, Register, RAM, ROM, Secondary storage, Optical storage, Magnetic storage, Solid state storage, Operating system **Disciplinary Reading -** Read an overview article on embedded systems and CPU architecture. Annotate with key terms. **Disciplinary Writing -** Write an explanation of the difference between RAM and cache memory.	**Key Vocab** Variable, Constant, Data type, Integer, String, Boolean, Input, Output, Selection, IF statement, Arithmetic operator, Logical operator, Random number, Trace table, Syntax error **Disciplinary Reading -** Read example code and identify key vocabulary used (e.g., constant, variable, input, output). **Disciplinary Writing -** Write pseudocode with inline comments describing each line.	**Key Vocab** Algorithm, Decomposition, Abstraction, Pseudocode, Flowchart, Input, Process, Output, Linear search, Binary search, Merge sort, Bubble sort, Iteration, Efficiency **Disciplinary Reading -** Read and compare merge and bubble sort explanations. Create a comparison table. **Disciplinary Writing -** Write a step-by-step explanation of how a binary search works.	**Key Vocab** Iteration, For loop, While loop, Nested loop, Selection, Data validation, Subroutine, Function, Procedure, Array, Index, Record, Parameter, Return value, Modular programming **Disciplinary Reading -** Read about arrays and subroutines in Python. Annotate examples. **Disciplinary Writing -** Create a glossary of terms used when working with loops and functions.	**Key Vocab** Binary, Denary, Hexadecimal, Bit, Byte, Kilobyte, Megabyte, Gigabyte, Terabyte, Binary arithmetic, Binary shift, Character set, ASCII, Unicode, Compression **Disciplinary Reading -** Read about binary arithmetic and shifts. Practice vocabulary in context. **Disciplinary Writing -** Write a worked example of converting a binary value to denary and explaining each step.	**Key Vocab** Network, LAN, WAN, PAN, Topology, Star topology, Bus topology, Wired, Wireless, Protocol, TCP/IP, Firewall, Malware, Phishing, Penetration testing **Disciplinary Reading -** Read about network security measures. Create flashcards for each term (e.g., firewall, malware, phishing). **Disciplinary Writing -** Write a scenario-based analysis of how a cyberattack might occur and be prevented.
Numeracy	• **Interpreting storage sizes and units** (e.g., KB, MB, GB, TB) and calculating capacity requirements. • **Comparing performance specifications** using numerical values (e.g., 3.2 GHz vs 4.0 GHz, 4 MB vs 8 MB cache). • **Drawing and reading logic diagrams**, using truth tables to show binary input/output results. • **Working with binary values and memory**	• **Arithmetic operations** – applying addition, subtraction, division, multiplication, modulus, and powers in programming contexts. • **Random number use** – working within ranges and understanding probability in simulations. • **Logical reasoning** – evaluating conditions using Boolean logic	• **Logical reasoning and sequencing** – following steps accurately in a defined order. • **Binary decision-making** – used in binary search and flowchart logic. • **Working with comparisons and loops** – understanding iteration and conditions used in sorting algorithms. • **Structuring trace tables** – to monitor variable changes	• **Counting and index manipulation** in loops and arrays. • **Using comparison and arithmetic operators** in validation and conditionals. • **Tracing changes in numeric values** through loops in trace tables. • **Understanding ranges and bounds** in arrays and loops (e.g., loop from 0 to n−1). • **Applying logic operators** in nested	• **Place value calculations** in base-2 and base-16 systems. • **Binary conversions** – between binary, denary, and hexadecimal. • **Binary addition** and **recognising overflow**. • **Using powers of 2** for shifts, conversions, and unit calculations. • **Calculating file sizes** using formulas (e.g., image width × height × colour depth).	• **Data transmission calculations** – potentially understanding bandwidth, data rate, or latency comparisons. • **Interpreting statistical data** – such as success/failure rates of different topologies or attack types.

	registers conceptually and numerically.	to determine program flow. • **Reading and constructing trace tables** – step-by-step tracking of variable changes and outputs, reinforcing sequencing and logic. • **Understanding data types** – recognising implications of using integers vs floats in numerical calculations.	through algorithm execution.	decision-making (e.g., AND/OR). • **Working with structured data** like records requires an understanding of grouping and accessing multiple data types logically.	• **Understanding ratios and percentages** when explaining compression. • **Working with base-10 multiples** (KB = 1,000 bytes or 1,024 depending on context).	
Personal Development	• N/A	• N/A	• **Coding Club** – Available for pupils to embed and enhance their programming skills at lunch time.	• **Coding Club** – Available for pupils to embed and enhance their programming skills at lunch time.	• N/A	• N/A

SUBJECT Curriculum Map

Year 11 – GCSE Computer Science

Carnforth High School

Rationale and Links to The National Curriculum
Studying GCSE Computer Science provides students with the knowledge and skills needed to understand and shape the digital world around them. Technology is at the core of modern life, from smartphones and apps to healthcare systems, transportation, and entertainment. By learning how computers work and how software is created, students gain insight into the foundations of innovation and problem-solving in the 21st century. The course develops both practical and analytical skills. Programming encourages logical thinking, creativity, and resilience, as students design, test, and improve solutions to real problems. Understanding computer systems, data, and networks also helps students become informed users and creators of technology, not just passive consumers. These skills are highly transferable, supporting success in mathematics, science, and many other disciplines. GCSE Computer Science is also an excellent preparation for future opportunities. With the growing demand for computing expertise in almost every industry, the subject opens pathways to further study and a wide range of careers, from software development and cybersecurity to engineering and data science. Most importantly, it empowers students to engage confidently with the digital world and to contribute positively to the technological future. Pupils are given the opportunity to study GCSE Computer Science in Year 10 and Year 11, which helps them to develop their capability, creativity and knowledge in computer science, develop and apply their analytic, problem-solving, design, and computational thinking skills, understand how changes in technology affect safety, including new ways to protect their online privacy and identity, and how to identify and report a range of concerns.

	Half Term 1 – Ethical, Legal and Environmental Impact (15 Lessons)	Half Term 2 – Programming Robust and Secure Programs (16 Lessons)	Half Term 3 – Databases (Algorithms) (14 Lessons)	Half Term 4 – GCSE Revision (12 Lessons)	Half Term 5 – GCSE Exams
Key Topics	• Explain a range of ethical, legal and environmental impacts of digital technology • Know the risks the technologies pose • Topics covered will be cyber security,	• Be able to write a validation routine and authentication routine • Know what testing means in the context of a program	• Understand the use of relational databases • Be able to utilise the key elements of a database; Tables, records, fields, data types, primary keys & foreign keys	• Revision and Preparation for Summer Exams – These lessons are split between paper 1 and paper 2 practice • Lessons give coverage of all topic	• Revision and Preparation for Summer Exams – Expected completion early May. • This time will be spent preparing for final exams. The lessons at this time will be reviewing all topics covered over the Computer Science course.

	mobile tech, wireless, cloud storage, hacking, wearable tech, computer-based implants, autonomous vehicles	• Understand what test data is and types of test data • They will know the different types of error that can occur in a computer program • Identify errors in an algorithm or program	• Understand the concept of data redundancy • Use SQL to retrieve data from a database • Use SQL to insert, edit and delete data in a database	areas, providing focus on weaknesses identified in mock performance	
Substantive Knowledge (The knowledge the students will develop)	• **Ethical issues**: Pupils know a range of **ethical impacts** of digital technology, such as: a) Privacy concerns (e.g. surveillance, data collection) b) Bias in algorithms and AI c) Social impacts of automation and job loss • **Legal impacts**: Pupils know the implications of: a) The Computer Misuse Act b) The Data Protection Act c) Copyright law and software licensing • **Environmental impacts**: Pupils know how digital technology affects the environment through: a) Energy consumption	• Pupils know what **validation routines** are and how they help ensure **input data is acceptable** (e.g. length, format, range). • Pupils know what **authentication routines** are and how they protect systems by **verifying user identity** (e.g. passwords, two-factor authentication). • Pupils know what **testing** means in programming — the process of checking that a program behaves as intended. • Pupils know the different **types of test data**: a) **Normal** (expected/typical values)	• Pupils know the purpose and advantages of **relational databases**, including how they reduce **data redundancy** and improve data integrity. • Pupils know the core components of a database: a) **Tables** – collections of related data b) **Records** – individual entries (rows) c) **Fields** – attributes of data (columns) d) **Data types** – such as text, number, date/time e) **Primary keys** – unique identifiers for each record f) **Foreign keys** – fields that link tables together • Pupils know what **data redundancy** is	• See previous units	

	b) E-waste and disposal c) Manufacturing and transport of tech • **Technology risks**: Pupils know risks posed by: a) Cyber security threats b) Cloud storage vulnerabilities c) Hacking and data interception • **Emerging technologies**: Pupils know the implications of: a) Wearable tech and implants b) Autonomous vehicles c) Mobile and wireless technologies	b) **Boundary** (values on the edge of acceptable input) c) **Erroneous/Invalid** (values that should be rejected) • Pupils know the types of **program errors**: a) **Syntax errors** – code that breaks language rules b) **Logic errors** – code that runs but doesn't behave as expected c) **Runtime errors** – errors that occur during execution (e.g. divide by zero) • Pupils know how to **identify and locate errors** in both **algorithms and written code**.	and why **normalisation** and relational links help prevent it. • Pupils know how to **write SQL (Structured Query Language)** commands to: a) **Retrieve data** (`SELECT`, `WHERE`, `ORDER BY`) b) **Insert data** (`INSERT INTO`) c) **Update data** (`UPDATE ... SET ...`) d) **Delete data** (`DELETE FROM`)		
Disciplinary Knowledge (The skills and approaches that students will develop)	• **Identify and explain real-world impacts** of digital technology decisions. • **Analyse ethical dilemmas** related to current technologies (e.g. surveillance vs safety). • **Evaluate pros and cons** of emerging technologies (e.g. implants for health vs data privacy).	• **Write code** that includes appropriate **input validation** (e.g. while loops checking input). • **Implement basic authentication routines** (e.g. username/password checking). • **Plan and carry out testing**, using suitable **test data**.	• **Design and structure a relational database**, with appropriately linked tables. • **Identify and use appropriate data types** for different fields. • **Apply primary and foreign keys** to ensure relational integrity. • **Use SQL commands** to:	• See previous units	

	• **Interpret legal requirements** and apply them to case studies or scenarios. • **Assess environmental consequences** of digital technologies, including carbon footprint and waste. • **Use critical thinking** to form balanced arguments about societal effects. • **Explain trade-offs** between convenience, cost, ethics, and security in tech use.	• **Select and justify** types of test data to use. • **Trace code** manually (e.g. using trace tables) to identify logic or runtime errors. • **Debug and fix errors** in pseudocode or actual code. • **Analyse error messages** and use them to guide corrections.	a) **Extract data** that meets given criteria b) **Sort or filter data** c) **Insert, update and delete** records effectively • **Analyse and improve database structure** to avoid redundancy and duplication. • **Interpret existing SQL queries** and explain their function.		
Assessment (The methods that teachers will use to assess the progress of all students)	• Mid-unit assessment paper (past exam questions) • End-of-unit assessment paper (past exam questions)	• November Mock Exam (Paper 1 and 2 combined) • End-of-unit assessment paper (past exam questions)	• Mid-unit assessment paper (past exam questions) • End-of-unit assessment paper (past exam questions)	• March Mock Exam (Paper 1 and 2 combined)	
Reading, Writing and Vocabulary	**Key Vocab** Ethics, Law, Environment, Cyber security, Cloud storage, Mobile technology, Wireless, Hacking, Wearable technology, Autonomous vehicles, Data protection, Digital divide, Sustainability, Intellectual property, Computer misuse	**Key Vocab** Validation, Authentication, Test plan, Test data, Normal data, Boundary data, Invalid data, Error, Syntax error, Logic error, Runtime error, Debugging, Maintainability, Comments, Robustness	**Key Vocab** Database, Relational database, Table, Record, Field, Data type, Primary key, Foreign key, Data redundancy, SQL, SELECT, INSERT, UPDATE, DELETE, Query **Disciplinary Reading -** Read an overview article on embedded systems	• See previous units	

	Disciplinary Reading - Read an overview article on embedded systems and CPU architecture. Annotate with key terms. **Disciplinary Writing -** Write an explanation of the difference between RAM and cache memory.	**Disciplinary Reading -** Read an overview article on embedded systems and CPU architecture. Annotate with key terms. **Disciplinary Writing -** Write an explanation of the difference between RAM and cache memory.	and CPU architecture. Annotate with key terms. **Disciplinary Writing -** Write an explanation of the difference between RAM and cache memory.		
Numeracy	• **Interpreting statistics** on energy usage, emissions, or e-waste growth. • **Comparing data usage or cost** (e.g., cloud storage vs physical media). • **Understanding percentages** (e.g., percentage of breaches due to weak passwords). • **Analysing trends** in tech adoption and its impact using graphs or reports.	• **Working with boundary values** and ranges in validation (e.g. ensuring age is between 0 and 120). • **Logical reasoning** when using AND/OR conditions in validation and authentication. • **Pattern recognition** in spotting repeated errors or logic faults. • **Understanding limits and constraints** numerically (e.g. password length, number ranges). • **Using trace tables**, which involves **sequential logical steps**, tabulating values, and predicting output.	• **Logical reasoning** in constructing and evaluating SQL queries (e.g. using `AND`, `OR`, `NOT`). • **Using comparison operators** (e.g. `>`, `<`, `=`, `BETWEEN`, `LIKE`) in `WHERE` clauses. • **Counting and aggregating data** (e.g. using SQL functions like `COUNT`, `AVG`, `SUM` if included).	• See previous units	
Personal Development	• **Computing Support** – Additional sessions to support revision for GCSE	• **Computing Support** – Additional sessions to support revision for GCSE	• **Coding Club** – Available for pupils to embed and enhance their	• **Coding Club** – Available for pupils to embed and enhance their	

			programming skills at lunch time. • **Computing Support** – Additional sessions to support revision for GCSE	programming skills at lunch time. • **Computing Support** – Additional sessions to support revision for GCSE	

SUBJECT Curriculum Map

Year 10 – Cambridge Nationals Creative iMedia

Rationale and Links to The National Curriculum
Studying Creative iMedia allows students to explore the exciting world of digital media and develop practical skills that are highly relevant in today's creative industries. From graphic design and animation to video production and interactive media, the course provides opportunities to design, create, and evaluate digital products used every day in advertising, entertainment, and communication. The subject is hands-on and encourages creativity while building technical expertise. Students learn to use industry-standard software and develop project management skills by working through realistic media briefs. This approach not only strengthens problem-solving and organisational abilities but also helps students understand how to meet client needs and engage target audiences effectively. Creative iMedia also equips students with valuable transferable skills such as teamwork, time management, and attention to detail. These are essential for success in education and employment alike. The qualification opens pathways into further study in creative and digital media, IT, and design, as well as careers in film, television, web development, marketing, and game design. Most importantly, the course empowers students to combine creativity with technology, giving them the confidence to produce original digital content and to participate fully in the fast-paced, media-driven world around them. Pupils are given the opportunity to study Cambridge Nationals Creative iMedia in Year 10 and Year 11, which helps them to develop their capability, creativity and knowledge in Digital Media and Information Technology.

	Half Term 1 – R093 Use of Media in Industry (15 Lessons)	**Half Term 2 – R093/R094 Planning and Design (17 Lessons)**	**Half Term 3 – R094 Graphics Editing Tools and Techniques (14 Lessons)**	**Half Term 4 – R094 Graphics Editing Tools and Techniques (12 Lessons)**	**Half Term 5 – R094 NEA Assessment (14 Lessons)**	**Half Term 6 – R099 Gaming Conventions and Features (14 Lessons)**
Key Topics	• Media industry sectors and products	• Work planning and documents used to support ideas generation	• Techniques to plan visual identity and digital graphics • Tools and techniques to create visual	• Technical skills to source, create and prepare assets for use within digital graphics	• NEA Assessment (Working on and submit for moderation)	• Introduction to digital games

	• How style, content and layout are linked to the purpose • Client requirements and how they are defined • Audience demographics and segmentation • Media codes used to convey meaning, create impact and/or engage audiences	• Documents used to design/plan media products • Purpose, features, elements and design of visual identity • Graphic design concepts and conventions • Properties of digital graphics and use of assets	identity and digital graphics • Technical skills to source, create and prepare assets for use within digital graphics	• Techniques to save and export visual identity and digital graphics • Distribution considerations and file formats • NEA Assessment (working on)		• Features and conventions of digital games • Creativity in digital games • Resources required to create digital games
Substantive Knowledge (The knowledge the students will develop)	• Pupils know the **different media industry sectors** (e.g., film, TV, gaming, publishing, web, advertising) and the products they create. • Pupils know how **style, content, and layout** are chosen and adapted to suit the **purpose** of a media product. • Pupils know what **client requirements** are and how they are clearly defined in a design brief. • Pupils know about **audience demographics** (e.g., age, gender, income, ethnicity) and **audience**	• Pupils know the different **work planning methods** (e.g., production schedules, task lists, Gantt charts) and their purpose in supporting ideas generation. • Pupils know the **documents used to design and plan media products** (e.g., mood boards, mind maps, storyboards, scripts, visualisation diagrams). • Pupils know the **purpose, features, and elements of visual identity** (e.g., logos, colour palettes, typography).	• Pupils know **techniques to plan visual identity and digital graphics**, including the use of design documents (e.g., mood boards, visualisation diagrams). • Pupils know the range of **tools and techniques** available in graphics software (e.g., layers, selection tools, cropping, adjustment tools, filters). • Pupils know the **technical requirements of digital assets** (file formats, resolution, compression, colour modes) and how they impact quality and usability.	• Pupils know the **technical skills to source, create, and prepare assets** for use in digital graphics, including ensuring quality and copyright compliance. • Pupils know **techniques to save and export digital graphics**, including file types suitable for different purposes (e.g., JPEG, PNG, GIF, PDF). • Pupils know **distribution considerations**, including appropriate file formats, resolution, and platform requirements.	• Pupils know the **requirements of the NEA (Non-Exam Assessment)**, including submission guidelines, deadlines, and moderation procedures. • Pupils know how to **apply planning, design, and technical skills** from previous units to complete a media product. • Pupils know the importance of **meeting client requirements, target audience needs, and assessment criteria**. • Pupils know how to **evaluate their own work** and make iterative improvements.	• Pupils know the **different types of digital games** (e.g., platform, shooter, puzzle, simulation, RPG) and their characteristics. • Pupils know the **features and conventions of digital games**, such as objectives, rules, levels, scoring, rewards, and player interaction. • Pupils know how **creativity is applied** in digital games, including narrative, character design, visual style, and sound. • Pupils know the **resources required to create digital games**, including

	segmentation methods. • Pupils know the role of **media codes** (technical, symbolic, written) in creating meaning, impact, and audience engagement.	• Pupils know the key **graphic design concepts and conventions** (e.g., alignment, contrast, proximity, balance, white space). • Pupils know the **properties of digital graphics** (e.g., resolution, file types, compression) and how to use and source **assets** appropriately.	• Pupils know how to **source, create, and prepare assets** in line with copyright law and project requirements.	• Pupils know the **requirements of the NEA (Non-Exam Assessment)** and how work should meet client and audience needs.		software tools, hardware, assets, and human resources (e.g., designers, programmers, artists).
Disciplinary Knowledge (The skills and approaches that students will develop)	• **Analyse media products**, identifying their sector, style, and purpose. • **Interpret client briefs** to extract requirements and success criteria. • **Match product features** to specific target audience demographics. • **Apply knowledge of media codes** when creating or critiquing media products. • **Evaluate media products** against client requirements and audience expectations.	• **Create and use planning documents** effectively to organise project work. • **Generate and present design ideas** using recognised conventions. • **Develop a visual identity** that reflects a client brief and audience needs. • **Apply design principles** to produce professional-looking media products. • **Select and prepare digital assets**, ensuring they are fit for purpose and legally sourced.	• **Plan and design digital graphics**, showing links between visual identity and client requirements. • **Use specialist graphics software** to create, edit, and enhance digital graphics. • **Apply tools and techniques** to combine text and images effectively. • **Organise and manage digital assets**, ensuring correct file types, sizes, and optimisation for purpose. • **Evaluate digital graphics**, considering audience, purpose,	• **Prepare assets** for use in graphics projects, ensuring correct resolution and format. • **Export files** in appropriate formats for print, web, or other media. • **Make informed decisions** about file types, considering quality, size, and distribution platform. • **Apply project management skills** to meet NEA assessment deadlines and criteria. • **Evaluate digital graphics** for technical accuracy, purpose, and audience suitability.	• **Plan and manage a project** independently, including workflow, milestones, and deadlines. • **Develop a final media product** using digital graphics, visual identity, and other media skills. • **Apply technical skills** to source, create, prepare, and export assets appropriately. • **Evaluate and refine** the media product, documenting decisions and improvements. • **Prepare and submit work** according to NEA moderation requirements.	• **Analyse digital games** to identify features, conventions, and design choices. • **Apply creative thinking** to propose or design game concepts. • **Plan resources** needed for a game project, including digital assets, tools, and personnel. • **Evaluate games** in terms of audience engagement, creativity, and adherence to conventions.

			and technical accuracy.			
Assessment (The methods that teachers will use to assess the progress of all students)	• Mid-unit assessment paper (past exam questions) • End-of-unit assessment paper (past exam questions)	• Mid-unit assessment paper (past exam questions) • End-of-unit assessment paper (past exam questions)	• Mid-unit assessment online • End-of-unit mock NEA review	• Mid-unit mock NEA review • NEA assessment review	• NEA assessment submission	• Mid-unit assessment online • End-of-unit assessment paper (past exam questions)
Reading, Writing and Vocabulary	**Key Vocab** Media sector, Media product, Style, Content, Layout, Purpose, Client requirements, Audience demographics, Segmentation, Target audience, Media codes, Technical codes, Symbolic codes, Written codes, Engagement **Disciplinary Reading -** Read a case study of a media campaign (e.g., a film release or advertising campaign). Highlight how style, content, and layout were chosen to meet the client's brief and engage the target audience. **Disciplinary Writing -** Write a short analysis of a chosen media product (poster, website, game, or advert), explaining how it reflects client	**Key Vocab** Work plan, Task list, Gantt chart, Mood board, Mind map, Storyboard, Visualisation diagram, Visual identity, Typography, Colour palette, Design convention, Alignment, Resolution, File type, Asset **Disciplinary Reading -** Read a design case study (e.g., the creation of a logo or advertising campaign). Highlight how planning documents and visual identity elements were used to shape the product. **Disciplinary Writing -** Write a design proposal for a new media product (e.g., a poster, app, or website), describing how you would use planning documents, visual	**Key Vocab** Visual identity, Mood board, Visualisation diagram, Layer, Selection tool, Adjustment tool, Cropping, Resolution, DPI (dots per inch) / PPI (pixels per inch), Compression, Colour mode (RGB, CMYK), Asset, File format, Optimisation **Disciplinary Reading -** Read a professional guide or tutorial on preparing images for print and web. Identify and define key terms such as resolution, compression, and optimisation. **Disciplinary Writing -** Write a step-by-step workflow explaining how you sourced, prepared, and edited assets for a digital graphic, using	**Key Vocab** Asset, Source, Create, Prepare, Export, File format, JPEG, PNG, GIF, PDF, Resolution, Compression, Distribution, NEA, Platform requirements **Disciplinary Reading -** Read a guide on exporting and optimising graphics for multiple platforms. Highlight key terms like compression, resolution, and file format. **Disciplinary Writing -** Write a technical report describing how you prepared, exported, and distributed digital graphics for your NEA project, using disciplinary vocabulary.	**Key Vocab** NEA, Moderation, Client requirements, Target audience, Planning, Workflow, Milestones, Media product, Asset, Export, Evaluation, Iteration, Refinement, Deadline, Submission **Disciplinary Reading -** Read a NEA brief and highlight how the workflow, planning, and evaluation are documented, noting key terms used. **Disciplinary Writing -** Write a reflective report on your NEA project, explaining how you applied planning, design, and technical skills to meet client and audience needs, using subject-specific vocabulary throughout.	**Key Vocab** Digital game, Genre, Features, Conventions, Objective, Rules, Level, Scoring, Reward, Player interaction, Narrative, Visual style, Sound, Asset, Resource **Disciplinary Reading -** Read a case study of a popular digital game. Highlight key features, conventions, and creative choices, noting how resources were used to create it. **Disciplinary Writing -** Write a short design report proposing a new digital game, describing its features, genre, creative elements, and the resources required, using disciplinary vocabulary.

	requirements, targets a specific demographic, and uses media codes effectively.	identity, and digital assets to meet a client's brief.	disciplinary vocabulary throughout.			
Numeracy	• **Interpreting audience data**, such as percentages in demographic breakdowns. • **Working with statistics** from surveys or market research. • **Using scales or proportions** when planning layouts. • **Analysing charts or graphs** showing audience segmentation or market share.	• **Using timelines and schedules** (e.g., sequencing tasks, estimating durations). • **Interpreting proportions and ratios** when designing layouts. • **Applying measurements** for resolution, image dimensions, and file sizes. • **Analysing graphical properties** such as	• **Measuring and adjusting resolution** (dpi/ppi, pixel dimensions). • **Scaling and resizing** images proportionally. • **Using grids, alignment, and symmetry** to ensure balanced design. • **Applying ratios and proportions** in layouts (e.g., rule of thirds).	• **Calculating file sizes** and compression ratios. • **Adjusting image resolution** for print vs digital display. • **Scaling images proportionally** while maintaining quality. • **Comparing storage requirements** for different file formats.	• **Estimating time and resources** to meet project deadlines. • **Scaling and resizing digital assets** to correct dimensions and resolution. • **Calculating file sizes** for distribution or submission. • **Applying ratios, proportions, and measurements** in layouts or visual design.	• **Scoring and points systems** (calculations for player rewards or progression). • **Level design measurements** (e.g., grid-based layouts or scaling objects). • **Resource management calculations** (time, budget, or number of assets needed). • **Probability and chance mechanics** in gameplay (e.g.,

		aspect ratios and pixel density.	• **Interpreting file sizes** and storage requirements.			random events or outcomes).
Personal Development	• N/A	• N/A	• **Graphics Club** – Available for pupils to embed and enhance their graphics editing skills at lunch time.	• **Graphics Club** – Available for pupils to embed and enhance their graphics editing skills at lunch time.	• N/A	• N/A

SUBJECT Curriculum Map

Year 11 – Cambridge Nationals Creative iMedia

Rationale and Links to The National Curriculum
Studying Creative iMedia allows students to explore the exciting world of digital media and develop practical skills that are highly relevant in today's creative industries. From graphic design and animation to video production and interactive media, the course provides opportunities to design, create, and evaluate digital products used every day in advertising, entertainment, and communication. The subject is hands-on and encourages creativity while building technical expertise. Students learn to use industry-standard software and develop project management skills by working through realistic media briefs. This approach not only strengthens problem-solving and organisational abilities but also helps students understand how to meet client needs and engage target audiences effectively. Creative iMedia also equips students with valuable transferable skills such as teamwork, time management, and attention to detail. These are essential for success in education and employment alike. The qualification opens pathways into further study in creative and digital media, IT, and design, as well as careers in film, television, web development, marketing, and game design. Most importantly, the course empowers students to combine creativity with technology, giving them the confidence to produce original digital content and to participate fully in the fast-paced, media-driven world around them. Pupils are given the opportunity to study Cambridge Nationals Creative iMedia in Year 10 and Year 11, which helps them to develop their capability, creativity and knowledge in Digital Media and Information Technology.

	Half Term 1 – R099 Pre-production planning (15 Lessons)	**Half Term 2 – R099 Digital Games Tools and Techniques (16 Lessons)**	**Half Term 3 – R099 Digital Games Tools and Techniques (14 Lessons)**	**Half Term 4 – R099 NEA (12 Lessons)**	**Half Term 5 – Exam**
Key Topics	• Pre-production and planning documentation and techniques for digital games	• Techniques to obtain, create and manage assets • Techniques used to create animation with digital games	• Techniques to save and export digital games • Techniques to test/check and	• NEA Assessment (Working on and submit for moderation)	• Revision and Preparation for Summer Exam – Expected completion early June. • Distribution platforms and media to reach audiences

			review animation with audio • Improvements and further developments		• Properties and formats of media files • Sources of research and types of research data • The legal issues that affect media • Job roles in the media industry • Revision and mock papers/tests
Substantive Knowledge (The knowledge the students will develop)	• Pupils know the **purpose of pre-production and planning** in digital game development. • Pupils know the **types of planning documentation** used, including: ○ **Storyboards** ○ **Mood boards** ○ **Mind maps** ○ **Visualisation diagrams** ○ **Game design documents** • Pupils know **techniques for planning** games, such as sequencing levels, defining objectives, designing characters, and specifying resources. • Pupils know how planning **links to client requirements, target audience, and gameplay experience**.	• Pupils know **techniques to obtain, create, and manage assets** for digital games, including sprites, backgrounds, audio, and interactive elements. • Pupils know the **principles of animation in digital games**, such as keyframes, frame rate, motion paths, and timing. • Pupils know how **animation techniques enhance gameplay**, including feedback, storytelling, and player engagement. • Pupils know the importance of **organising and optimising assets** for use in game development.	• Pupils know **techniques to save and export digital games**, including file formats, platforms, and compatibility considerations. • Pupils know **methods to test, check, and review** animation and audio integration within digital games. • Pupils know how to identify **improvements and further developments** to enhance gameplay, user experience, or technical performance. • Pupils know the importance of **iteration** in refining digital games based on feedback and testing results.	• Pupils know the **requirements of the NEA (Non-Exam Assessment)**, including submission guidelines, deadlines, and moderation procedures. • Pupils know how to **apply planning, design, and technical skills** from previous units to complete a media product. • Pupils know the importance of **meeting client requirements, target audience needs, and assessment criteria**. • Pupils know how to **evaluate their own work** and make iterative improvements.	
Disciplinary Knowledge (The skills and	• **Create pre-production**	• **Source, create, and manage digital**	• **Save and export games** appropriately for different	• **Plan and manage a project** independently,	

approaches that students will develop)	**documentation** for a digital game project. • **Plan game structure and mechanics** to meet design objectives. • **Use design documents** to communicate ideas clearly to a team or client. • **Evaluate planning effectiveness**, ensuring all requirements and gameplay considerations are addressed.	**assets** effectively for a game project. • **Apply animation techniques** to characters, objects, or backgrounds in a game. • **Integrate assets and animations** into a game workflow. • **Evaluate the effectiveness** of animations and assets in contributing to gameplay and audience engagement.	platforms and file types. • **Test and review** game functionality, including animation, audio, and interactivity. • **Identify and implement improvements** to enhance user experience, gameplay, or technical performance. • **Document changes and development decisions** for iterative improvement. • **Evaluate the effectiveness** of enhancements and provide recommendations.	including workflow, milestones, and deadlines. • **Develop a final media product** using digital graphics, visual identity, and other media skills. • **Apply technical skills** to source, create, prepare, and export assets appropriately. • **Evaluate and refine** the media product, documenting decisions and improvements. • **Prepare and submit work** according to NEA moderation requirements.	
Assessment (The methods that teachers will use to assess the progress of all students)	• Mid-unit assessment online • End-of-unit assessment online	• Mid-unit assessment online • End-of-unit assessment online	• Mid-unit mock NEA review • NEA assessment review	• NEA assessment submission	
Reading, Writing and Vocabulary	**Key Vocab** Pre-production, Planning, Documentation, Storyboard, Mood board, Mind map, Visualisation diagram, Game design document, Client	**Key Vocab** Asset, Sprite, Background, Audio, Keyframe, Frame rate, Motion path, Timing, Animation, Game mechanics, Integration,	**Key Vocab** Save, Export, File format, Platform, Compatibility, Test, Review, Animation, Audio, Integration, Iteration, Improvement, Development, User	**Key Vocab** NEA, Moderation, Client requirements, Target audience, Planning, Workflow, Milestones, Media product, Asset, Export, Evaluation,	

	requirements, Target audience, Gameplay mechanics, Level design, Asset, Sequencing, Resources **Disciplinary Reading -** Read a case study or article on the pre-production process for a digital game. Highlight and define key terms like storyboard, visualisation diagram, and gameplay mechanics. Annotate with key terms. **Disciplinary Writing -** Write a pre-production plan for a proposed digital game, describing the planned levels, assets, and gameplay mechanics, using disciplinary vocabulary throughout.	Optimisation, Interactivity, Workflow, Feedback **Disciplinary Reading -** Read a tutorial or case study on animation in digital games. Highlight key terms such as keyframe, frame rate, motion path, and sprite. **Disciplinary Writing -** Write a technical guide explaining how you sourced and animated assets for your digital game project, using disciplinary vocabulary throughout.	experience (UX), Gameplay **Disciplinary Reading -** Read a tutorial or case study on testing and exporting digital games. Highlight key terms such as iteration, compatibility, and integration. **Disciplinary Writing -** Write a review report on a digital game you created, explaining how you tested and refined animation and audio, and what improvements you implemented, using disciplinary vocabulary.	Iteration, Refinement, Deadline, Submission **Disciplinary Reading -** Read a NEA brief and highlight how the workflow, planning, and evaluation are documented, noting key terms used. **Disciplinary Writing -** Write a reflective report on your NEA project, explaining how you applied planning, design, and technical skills to meet client and audience needs, using subject-specific vocabulary throughout.	
Numeracy	• **Sequencing levels and events** numerically to ensure logical progression. • **Timing and pacing calculations** for gameplay (e.g., duration of levels, animations). • **Applying grid systems and coordinates** in level design.	• **Calculating frame rates** and timing for smooth animation. • **Measuring object movement** and positions in coordinates or grid systems. • **Scaling and resizing assets** proportionally for consistency.	• **Calculating file sizes** for game assets and exported projects. • **Estimating performance impact** of added assets or features. • **Using coordinates and scaling** for animation adjustments.	• **Estimating time and resources** to meet project deadlines. • **Scaling and resizing digital assets** to correct dimensions and resolution. • **Calculating file sizes** for distribution or submission. • **Applying ratios, proportions, and measurements** in layouts or visual design.	

Personal Development	• N/A	• **Digital Games Club** – Pupils can practice their skills in game development	• N/A	• N/A	