Computer Science

Curriculum Intent and Long Term Plan – 2024-25

Subject Intent Statement

In computer science we believe in nurturing passion in developing useful knowledge that can be applied in adult life. We deliver a varied, high challenge and interesting curriculum that we assess consistently to foster confidence and pride from our students. It is vital that in this digital age, students are equipped with the essential knowledge required in our modern lives.

As Staff: We are always looking for the next improvement whether that be through innovation or by taking new risks. Like the world of computing, we never wish to stagnate. We are continually reflecting as staff in collaboration with our students to maximise the impact of our curriculum. Ultimately in computing, we make everything student centric and about all else, our primary goal is to make the students' experience a great one

As Students: You relish problem solving. Your learning does not stop when the bell does and you will go beyond the classroom learning in your free time. You love to design, develop and code solutions and are passionate, motivated and up for a challenge. You do not give up easily but know when to communicate with others to seek support to complex problems when you have tried your hardest to solve it yourself. Ultimately you arrive to each lesson ready to give 100% and you do it all with a smile.

Key Concepts

Key Stage 3	Key Stage 4	Key Stage 5
Problem Solving Algorithms Software and Hardware Logic and Binary Imedia concepts Online safety	Problem Solving Algorithms Software and Hardware Logic and Binary Imedia concepts Networks Impact of Computer Science	Problem Solving Algorithms Software and Hardware Logic and Binary Imedia concepts Networks Impact of Computer Science OOL and procedural coding Computational Thinking

Key Language/Terminology

Key Stage 3	Key Stage 4	Key Stage 5
Sequence, selection, iteration Variable Manipulation Target Audience AND, OR, NOT Vectors vs bitmap Netiquette	All from KS3 and: Protocols (TCP/IP etc..) Ethic, moral, cultural, environmental Translators – interpreter/compiler Defensive design CPU architecture	All from KS4 and: System design Layers (protocol) Boolean algebra/simplification Big O notation Databases (Inc ERD, normalisation) Web techs (HTML, CSS, Javascript) Memory management Data structures

Year 7

Curriculum Coherence

Y7 curriculum builds upon the different experiences' students will have had in Y6 from our feeder schools. This will include some experience in basic problem solving and object-based programming at an entry level.

Medium Term Plan Title/Topic	Themes/Concepts	Key Core Knowledge Foci	Application/Skills Foci	Ambitious Tier 2/3 Vocabulary	Assessment	Independent Learning
Expectations and character	Conduct within a computer room setting and general expectations for the course	Core virtues from school policy	Application of the expectations set in all lessons	Virtue specific vocab (eg moral, civic, etc..)	N/A	N/A
Toxicode	Problem solving	Introduction to simple coding concepts	Following algorithms	Sequence, Iterate	Formative – progress on platform	Can be completed in home setting
Core applications (office, onedrive, teams)	Orientation of core software used across the school	Functional knowledge and skills in using core applications	Functional knowledge and skills in using core applications	Folders, OneDrive, Cloud storage	N/A	
Code.org platform	Coding and problem solving via Object orientated approach	Basic programming concepts such as sequence, selection	Problem solving various tasks of varying difficulty	Algorithm, programming constructs	Formative via LessonUp Summative testing at end of half term	ILT will be set via platform to complete
Online Safety	Digital learning aspect on esafety	Grooming, dangers of being online	Effective use of online platforms to stay safe and report suspicious behaviour	Grooming, esafety	N/A	N/A

Logo Design	Basic logo design via Canva	Characteristics of effective logos	Effective logo design – specific tools in software	Vectors, pathing, audience targeting	Peer assessment of logos	ILT on logo design
Binary basics	Binary conversions, addition and subtraction	Binary conversions and why binary	Performing conversions	Binary, conversion, bit, denary	Summative binary test	Practice via nibblz.co.uk platform
Photo manipulation	Creating effective restaurant menu			Crop, transform, layers, rasterise, manipulation	Peer assessment hats activity	Potential ILT on Photopea
Hardware and Software	Basic input/output/storage And apps vs utilities	Devices that fit into each hardware group. Characteristics of apps and utilities	Identifying key components and their classification	Primary memory, input, output, storage, USB, Application , utility software, operating system	Summative testing	Revision via nibblz.co.uk or BBC bitesize
Turtle Academy	Problem solving, writing algorithms	Sequence, iteration coding	Solving set challenges using text-based toolkit	Procedure, FD, BK, RT, LT, PU, PD, Iterations	Summative code test	ILT on using Turtleacademy to create multifunction program
Trusting Content	Fake news and cross checking sources	Knowledge in internet sourcing and effective searching. Fake news	Searching and cross checking	Fake News, cross checking, trustful content (sources)	N/A	N/A
Logic Gates	AND OR and NOT gates	All major gates and truth tables	Working out truth tables for complex gate problems	AND, OR, NOT, Boolean, Algebra	Summative test	
Python	Python basics (sequence, selection)	Outputs, inputs, variables, IF/ELSE	Solving set challenges with learnt syntax	Syntax, IDE, runtime shell, variables, construct	Summative test	ILT on python principles
Microbits	OOP based programming	Sequencing, iteration. Real world device use	Solving set challenges using microbit ecosystem	Algorithms, sequencing, microprocessor, LED array	Summative test	
Web design	Audience targeting, website structure	Key components to making websites	Effective navigation and targeting audience.	Navigation bar, hyperlink, hypertext, Audience	Peer assessment	ILT on webdesign

Year 8

Curriculum Coherence

Y8 builds upon the knowledge and skill delivered in Y7 but takes them to more advanced levels. Greater levels of coding competence is expected and new platforms like scratch and vectr are introduced to solve more difficult tasks in problem solving, programming and creative media design.

Medium Term Plan Title/Topic	Themes/Concepts	Key Core Knowledge Foci	Application/Skills Foci	Ambitious Tier 2/3 Vocabulary	Assessment	Independent Learning
Expectations and character	Conduct within a computer room setting and general expectations for the course	Core virtues from school policy	Application of the expectations set in all lessons	Virtue specific vocab (eg moral, civic, etc..)	N/A	N/A
Toxicode	Problem solving – a refresh	Introduction to simple coding concepts	Following algorithms	Sequence, Iterate	Formative – progress on platform	Can be completed in home setting
Core applications (office, onedrive, teams)	Orientation of core software used across the school	Functional knowledge and skills in using core applications	Functional knowledge and skills in using core applications	Folders, OneDrive, Cloud storage	N/A	
Code.org platform	Coding and problem solving via Object orientated approach	Basic programming concepts such as sequence, selection	Problem solving various tasks of varying difficulty	Algorithm, programming constructs	Formative via LessonUp Summative testing at end of half term	ILT will be set via platform to complete
Netiquette	Digital literacy element	Online behaviour. Communications and their use	Appropriate online activity. Reporting content. Reflection on own use.	Netiquette, Trolling, griefing	N/A	N/A

Imedia – animated advert	Use of canva to create animated banner advert	Audience targeting, appropriate animation tools	Effective application of design tools to create adverts with subtle animation	Animation, transition	Peer assessment and refinement	Potential extension tasks available via canva platform relating to origin task
Games design	Pillars of game design and narrative design	Gaming pillars, Choose your own adventure style story delivery	Using Twinery to create text based adventure design	Lore, Narrative, Story Arc	Peer assessment by "playing" designed twines	Students will be able to extend Story driven games via twinery online from home
Imedia – music video	Video editing to set music	Inserting music and video. Trimming and transitioning. Text insertion. Effects	Effective trimming and timing of video elements to sound events. Effective use of overlay text	Transitions, layering, blending, trimming, splitting	Peer and Teacher based assessment of final video	ILT on creating music video for own topic focus.
Scratch project	Block based coding. Sequencing, selection and iteration	Coding using block based interface to solve problems of varying difficulty, ending with project.	Creation or a maze based game using acquired knowledge in skill lessons.	Stage, sprite, object, orientation, collision detection	Peer and Teacher assessed	ILT on creating own Scratch based game (not maze based)
Imedia – Photoshop artefact	Professional poster creation (Movie poster)	Using photopea tools – background removal, layering, blending, recolouring, text	Appropriate use of tools to target set audience and create appropriate and effective poster	Hue, saturation, transform, rotation, lasso, layering, exporting	Peer assessment and refinement opportunity.	ILT on creating photopea artefact

Year 9

Curriculum Coherence

Y9 builds upon the knowledge and skill delivered in Y7 and Y8 but takes them to more advanced levels. Problem solving difficulty increases as we spend more time in text based coding environments. We also extend our Imedia client briefs to give a more real world impression of how these skills are used in the outside world. Ultimately we are trying to best inform them in the differences between CS and Imedia so they can make suitable pathway choices going into GCSE.

Medium Term Plan Title/Topic	Themes/Concepts	Key Core Knowledge Foci	Application/Skills Foci	Ambitious Tier 2/3 Vocabulary	Assessment	Independent Learning
Expectations and character	Conduct within a computer room setting and general expectations for the course	Core virtues from school policy	Application of the expectations set in all lessons	Virtue specific vocab (eg moral, civic, etc..)	N/A	N/A
Toxicode	Problem solving – a refresh	Introduction to simple coding concepts	Following algorithms	Sequence, Iterate	Formative – progress on platform	Can be completed in home setting
Core applications (office, onedrive, teams)	Orientation of core software used across the school	Functional knowledge and skills in using core applications	Functional knowledge and skills in using core applications	Folders, OneDrive, Cloud storage	N/A	ILT on coding via code.org
Pathways session – IM vs CS	Fundamentals of Imedia vs Compsci as a pathway choice	Course make up of both subjects. Characteristics of ideal students for either pathway	Reflection on IM and CS to date. Careers links		N/A	Independent Research into either pathway (careers)
Sharing Media	Digital literacy element	Risks of sharing media online	Appropriate online activity. Reporting.		N/A	N/A
Personality Testing	Use of 16 Personalities testing to establish which personality students fall within.	Aspects linking to Introvert/Extrovert, Feeling/Thinking, Sensing/Intuitives and Judgers/Perceivers	Reflection on testing results against own beliefs of strengths/weaknesses. Investigating career pathways based upon personality	Introvert, extrovert, sensing, feeling, judging, intuitive	N/A	Deeper reading of personality profile information.
Imedia project – Consoles	Advanced logo design, animated adverts	Advanced Canva tools.	Effective audience targeting. Reflection on assets and refinement	Unify tool, punch tool, vectors	Peer assessment	ILT opportunity with additional graphical artefacts

Scratch mini project	Refresh of block based building. Project work	Use of broadcasts, conditionals, variables and iteration to create a game	Acquired skills lead into holiday break so ILT extension task will be set and assessment will be on return	Broadcasts, conditionals, iteration, collision detection	Peer and teacher assessed	ILT over half term break to adapt Scratch projects and complete them.
Photoshop Artefacts	Imedia mini project on the advanced use of photoshop	Use of advanced photoshop tools/skills to create professional assets fit for specific audiences	Application of skills to target audiences effectively	Colour replacement. JPG, PNG	Peer and Self assessed	ILT opportunities in artefact creation work via photopea.com
Minecraft EDU	Use of the Agent to complete set tasks – problem solving	Block based coding vs text based coding to program agent All core programming constructs	Application of commands to accomplish set goals	Agent, text based, object based	Teacher assessed	
PRIMM approach – Python	Learning text based language python – includes refresh and advancement in skills	Functions/procedures While/For/If – elif – else	Use python to solve set challenges	Prediction, Run, Investigation, Modify, Make	Self Assessed	Python Principles progression
Website Design	Imedia mini project in web design via Canva	Creation of website assets. Using navigation.	Implement assets to create effective websites that target specific audiences	Asset, Navbar	Peer Assessed	ILT on web design
Text representation	Binary representation of text	ASCII, UNICODE	Code cracking via binary codes Conversions between binary and text	ASCII, UNICODE, binary, bit	Self assessed	
Community working via MCEDU	Teamwork, leadership, working to deadlines in set environment	Using MCEDU – teams compete to complete set goals.	To effectively lead or be a part of a team in accomplishing set goals.	Community	Teacher assessed	

Year 10

Curriculum Coherence

Y10 GCSE Computing continues the journey by reinforcing the skills and knowledge gained in KS3 and investigating new lines of the curriculum specific to the OCR syllabus. These include CPUs, Memory, Networks, Software, Algorithms, Logic and binary to mention a few. Theory coverage is intersected by weekly coding development. Y10 delivery aims to cover at least 80% of all theory to ensure adequate time can be spent in recapping, revising and removing misconceptions during Y11.

Medium Term Plan Title/Topic	Themes/Concepts	Key Core Knowledge Foci	Application/Skills Foci	Ambitious Tier 2/3 Vocabulary	Assessment	Independent Learning
Expectations and character	Conduct within a computer room setting and general expectations for the course	Core virtues from school policy	Application of the expectations set in all lessons	Virtue specific vocab (eg moral, civic, etc..)	N/A	N/A
1.1 Systems Architecture	Purpose and function of the CPU and its performance	FDE cycle Registers Von Neumann CPU performance Embedded Systems	Explain FDE cycle in context Identify key registers Describe ways to improve CPU performance Identify examples of embedded systems		Both Formative elements via LessonUp delivery system and regular summative testing. Both self and peer assessed elements via SmartRevise will be used as well.	SmartRevise self-guided revision – at least 2x20 min sessions per week on agreed focus topic Self-guided coding progression
1.2 Memory and Storage	Primary, Secondary, Units,Binary and representation methods	RAM/ROM, Virtual memory Optical, magnetic, solid state Characteristics of storage Units (bits to petabytes) Number rep and conversions Text representation	Describe purpose or RAM/ROM Identify storage devices Explain the use of one storage device over another Convert binary/denary/hex Describe ASCII vs UNICODE	RAM/ROM, VM, optical, magnetic, solid state, bits, kilo, mega, peta, tera, binary, denary, hexadecimal, sampling, bit depth, pixels, dimensions, meta	Both Formative elements via LessonUp delivery system and regular summative testing. Both self and peer assessed elements via SmartRevise will be used as well.	SmartRevise self-guided revision – at least 2x20 min sessions per week on agreed focus topic Self-guided coding progression

		Image representation Sound representation Compression	Explain how sound is recorded Why do why compress data	data, ASCII, UNICODE, compression, lossy, lossless		
1.3 Computer Networks and Protocols	Networks, LAN s WANs Topologies Hardware P2P vs Client server DNS	Switches, Hubs, ethernet, routers, transmission media, DNS, hosting, the cloud, servers and clients, Star and Mesh	How is network performance affected? Describe hardware required to create a network Advantages of fibre optic over copper based media Adv/disad or cloud storage Draw mesh/star topology for given scenario	LAN/WAN, Internet, topology, P2P Switches, Hubs, ethernet, routers, transmission media, DNS, hosting, the cloud, servers and clients, Star and Mesh	Both Formative elements via LessonUp delivery system and regular summative testing. Both self and peer assessed elements via SmartRevise will be used as well.	SmartRevise self-guided revision – at least 2x20 min sessions per week on agreed focus topic Self-guided coding progression
1.4 Network Security	Threats to a computer system Identifying and preventing vulnerabilities	Forms of attack: Malware, social engineering, brute force, DOS, Data interception and theft SQL injection attack Prevention: Pentesting Anti-malware Firewalls, passwords, encryption, Physical secruity	Describe the forms of attack that can affect a network. Suggest suitable preventions and counters to common forms of attack. Identify three forms of physical security.	Malware, social engineering, DOS attack, SQL injection, pentesting, firewall, password, encryption	Both Formative elements via LessonUp delivery system and regular summative testing. Both self and peer assessed elements via SmartRevise will be used as well.	SmartRevise self-guided revision – at least 2x20 min sessions per week on agreed focus topic Self-guided coding progression
1.5 Operating Systems	Operating Systems Utility Software	UI, memory management, multitasking, peripherals, Users and file management Encryption, defrag and compression	Describe the purpose of an OS Explain two jobs the OS manages Describe the benefits of defragmentation	UI, multitask, peripheral, encryption, defragmentation, compression	Both Formative elements via LessonUp delivery system and regular summative testing. Both self and peer assessed elements via SmartRevise will be used as well.	SmartRevise self-guided revision – at least 2x20 min sessions per week on agreed focus topic Self-guided coding progression
1.6 ECEL impacts of Computers	Impact of digital technology Computer legislation	Ethical, legal, cultural, environmental and privacy issues DPA, CMA, Copyright designs and Patents and software licences	Discuss the impact of computers on varying real world scenarios Describe aspects and purpose of the specific legal acts.	Ethical, legal, cultural, environmental, DPA, CMA, Copyright designs and Patents and software licences	Both Formative elements via LessonUp delivery system and regular summative testing. Both self and peer assessed elements via SmartRevise will be used as well.	SmartRevise self-guided revision – at least 2x20 min sessions per week on agreed focus topic Self-guided coding progression
2.1 Algorithms	Computational Thinking Designing, creating and refining algorithms Searching and sorting	Abstraction, decomposition, algorithmic thinking Structure diagrams, pseudocode, flowcharts, common errors trace tables Binary and linear searcher Bubble, insertion and merge sort	Describe the purpose of abstraction. Draw a flowchart for a given problem Complete a trace table for a set algorithm Identify errors in provided code Perform various sorts on data	Abstraction, decomposition, algorithmic thinking Structure diagrams, pseudocode, flowcharts, common errors trace tables Binary and linear searcher Bubble, insertion and merge sort	Both Formative elements via LessonUp delivery system and regular summative testing. Both self and peer assessed elements via SmartRevise will be used as well.	SmartRevise self-guided revision – at least 2x20 min sessions per week on agreed focus topic Self-guided coding progression
2.2 Programming Techniques	Programming fundamentals Data types Additional techniques	Variables, constants, inputs, outputs and assignments Sequence, selection and iteration Operators Integer, real, Boolean, string and casting	Describe the role of a variable in a program Code in sequence, selection and iteration Apply appropriate operators Identify suitable data types to specific data	Variables, constants, inputs, outputs and assignments Sequence, selection and iteration Operators Integer, real, Boolean, string and casting	Both Formative elements via LessonUp delivery system and regular summative testing. Both self and peer assessed elements via SmartRevise will be used as well.	SmartRevise self-guided revision – at least 2x20 min sessions per week on agreed focus topic Self-guided coding progression

		File handling, SQL, arrays, sub programs. RNG	Write SQL to search for specific assets Open, read, write and close text files	File handling, SQL, arrays, sub programs. RNG		
2.3 Robust Programs	Defensive Design Testing	Authentication Validation Maintainability – comments, indentation, naming conventions, subprograms Purpose of testing Iterative/final testing Syntax/logic errors Test data: - Normal - Boundary - Invalid - erroneous	Identify 3 validation checks Describe two ways to maintain your code Give examples of the types of test data you would use and why Identify errors in code	Authentication Validation Maintainability – comments, indentation, naming conventions, subprograms Purpose of testing Iterative/final testing Syntax/logic errors Test data: - Normal - Boundary - Invalid	Both Formative elements via LessonUp delivery system and regular summative testing. Both self and peer assessed elements via SmartRevise will be used as well.	SmartRevise self-guided revision – at least 2x20 min sessions per week on agreed focus topic Self-guided coding progression
2.4 Boolean Logic	Logic gates Truth tables	AND, OR and NOT gates and their truth tables	Draw specific logic gates Draw multiple gates and work out the truth tables	AND, OR, NOT, logic gate, truth table, Boolean algebra	Both Formative elements via LessonUp delivery system and regular summative testing. Both self and peer assessed elements via SmartRevise will be used as well.	SmartRevise self-guided revision – at least 2x20 min sessions per week on agreed focus topic Self-guided coding progression
2.5 Translators and languages	Languages Integrated Development Environment (IDE)	High and low level languages Purpose of translators Compiler vs interpreter IDE editors, error diagnostics, runtime environment and translators	Compare low vs high languages Describe purpose of translators Compare compiler vs interpreter Describe the features of an IDE	Translator, compiler, interpreter, IDE, runtime	Both Formative elements via LessonUp delivery system and regular summative testing. Both self and peer assessed elements via SmartRevise will be used as well.	SmartRevise self-guided revision – at least 2x20 min sessions per week on agreed focus topic Self-guided coding progression

Year 11	**Curriculum Coherence**
	Y11 Starts by completing any untouched content from the syllabus and then runs highly interactive refresh sessions on the core topics. These are all delivered through our interactive teaching platform and offer insight into any misconceptions or areas for focus. Workshops are delivered on an ad-hoc basis in relation to student needs. Coding development continues to be a weekly exercise. Students are also given some self-guided learning time to focus on their priorities and are encouraged to so this either independently or in study groups/pairs

Medium Term Plan Title/Topic	Themes/Concepts	Key Core Knowledge Foci	Application/Skills Foci	Ambitious Tier 2/3 Vocabulary	Assessment	Independent Learning
Expectations and character	Conduct within a computer room setting and general expectations for the course	Core virtues from school policy	Application of the expectations set in all lessons	Virtue specific vocab (eg moral, civic, etc..)	N/A	N/A
Interactive revision 1.1	Purpose and function of the CPU and its performance	FDE cycle Registers Von Neumann CPU performance Embedded Systems	Explain FDE cycle in context Identify key registers Describe ways to improve CPU performance Identify examples of embedded systems		Both Formative elements via LessonUp delivery system and regular summative testing. Both self and peer assessed elements via SmartRevise will be used as well.	SmartRevise self-guided revision – at least 2x20 min sessions per week on agreed focus topic Self-guided coding progression

Interactive revision 1.2	Primary, Secondary, Units,Binary and representation methods	RAM/ROM, Virtual memory Optical, magnetic, solid state Characteristics of storage Units (bits to petabytes) Number rep and conversions Text representation Image representation Sound representation Compression	Describe purpose or RAM/ROM Identify storage devices Explain the use of one storage device over another Convert binary/denary/hex Describe ASCII vs UNICODE Explain how sound is recorded Why do why compress data	RAM/ROM, VM, optical, magnetic, solid state, bits, kilo, mega, peta, tera, binary, denary, hexadecimal, sampling, bit depth, pixels, dimensions, meta data, ASCII, UNICODE, compression, lossy, lossless	Both Formative elements via LessonUp delivery system and regular summative testing. Both self and peer assessed elements via SmartRevise will be used as well.	SmartRevise self-guided revision – at least 2x20 min sessions per week on agreed focus topic Self-guided coding progression
Interactive revision 1.3	Networks, LAN s WANs Topologies Hardware P2P vs Client server DNS	Switches, Hubs, ethernet, routers, transmission media, DNS, hosting, the cloud, servers and clients, Star and Mesh	How is network performance affected? Describe hardware required to create a network Advantages of fibre optic over copper based media Adv/disad or cloud storage Draw mesh/star topology for given scenario	LAN/WAN, Internet, topology, P2P Switches, Hubs, ethernet, routers, transmission media, DNS, hosting, the cloud, servers and clients, Star and Mesh	Both Formative elements via LessonUp delivery system and regular summative testing. Both self and peer assessed elements via SmartRevise will be used as well.	SmartRevise self-guided revision – at least 2x20 min sessions per week on agreed focus topic Self-guided coding progression
Interactive revision 1.4	Threats to a computer system Identifying and preventing vulnerabilities	Forms of attack: Malware, social engineering, brute force, DOS, Data interception and theft SQL injection attack Prevention: Pentesting Anti-malware Firewalls, passwords, encryption, Physical secruity	Describe the forms of attack that can affect a network. Suggest suitable preventions and counters to common forms of attack. Identify three forms of physical security.	Malware, social engineering, DOS attack, SQL injection, pentesting, firewall, password, encryption	Both Formative elements via LessonUp delivery system and regular summative testing. Both self and peer assessed elements via SmartRevise will be used as well.	SmartRevise self-guided revision – at least 2x20 min sessions per week on agreed focus topic Self-guided coding progression
Interactive revision 1.5	Operating Systems Utility Software	UI, memory management, multitasking, peripherals, Users and file management Encryption, defrag and compression	Describe the purpose of an OS Explain two jobs the OS manages Describe the benefits of defragmentation	UI, multitask, peripheral, encryption, defragmentation, compression	Both Formative elements via LessonUp delivery system and regular summative testing. Both self and peer assessed elements via SmartRevise will be used as well.	SmartRevise self-guided revision – at least 2x20 min sessions per week on agreed focus topic Self-guided coding progression
Interactive revision 1.6	Impact of digital technology Computer legislation	Ethical, legal, cultural, environmental and privacy issues DPA, CMA, Copyright designs and Patents and software licences	Discuss the impact of computers on varying real world scenarios Describe aspects and purpose of the specific legal acts.	Ethical, legal, cultural, environmental, DPA, CMA, Copyright designs and Patents and software licences	Both Formative elements via LessonUp delivery system and regular summative testing. Both self and peer assessed elements via SmartRevise will be used as well.	SmartRevise self-guided revision – at least 2x20 min sessions per week on agreed focus topic Self-guided coding progression
Interactive revision 2.1	Computational Thinking Designing, creating and refining algorithms Searching and sorting	Abstraction, decomposition, algorithmic thinking Structure diagrams, pseudocode, flowcharts, common errors trace tables Binary and linear searcher Bubble, insertion and merge sort	Describe the purpose of abstraction. Draw a flowchart for a given problem Complete a trace table for a set algorithm Identify errors in provided code Perform various sorts on data	Abstraction, decomposition, algorithmic thinking Structure diagrams, pseudocode, flowcharts, common errors trace tables Binary and linear searcher Bubble, insertion and merge sort	Both Formative elements via LessonUp delivery system and regular summative testing. Both self and peer assessed elements via SmartRevise will be used as well.	SmartRevise self-guided revision – at least 2x20 min sessions per week on agreed focus topic Self-guided coding progression

Interactive revision 2.2	Programming fundamentals Data types Additional techniques	Variables, constants, inputs, outputs and assignments Sequence, selection and iteration Operators Integer, real, Boolean, string and casting File handling, SQL, arrays, sub programs. RNG	Describe the role of a variable in a program Code in sequence, selection and iteration Apply appropriate operators Identify suitable data types to specific data Write SQL to search for specific assets Open, read, write and close text files	Variables, constants, inputs, outputs and assignments Sequence, selection and iteration Operators Integer, real, Boolean, string and casting File handling, SQL, arrays, sub programs. RNG	Both Formative elements via LessonUp delivery system and regular summative testing. Both self and peer assessed elements via SmartRevise will be used as well.	SmartRevise self-guided revision – at least 2x20 min sessions per week on agreed focus topic Self-guided coding progression
Interactive revision 2.3	Defensive Design Testing	Authentication Validation Maintainability – comments, indentation, naming conventions, subprograms Purpose of testing Iterative/final testing Syntax/logic errors Test data: - Normal - Boundary - Invalid erroneous	Identify 3 validation checks Describe two ways to maintain your code Give examples of the types of test data you would use and why Identify errors in code	Authentication Validation Maintainability – comments, indentation, naming conventions, subprograms Purpose of testing Iterative/final testing Syntax/logic errors Test data: - Normal - Boundary - Invalid	Both Formative elements via LessonUp delivery system and regular summative testing. Both self and peer assessed elements via SmartRevise will be used as well.	SmartRevise self-guided revision – at least 2x20 min sessions per week on agreed focus topic Self-guided coding progression
Interactive revision 2.4	Logic gates Truth tables	AND, OR and NOT gates and their truth tables	Draw specific logic gates Draw multiple gates and work out the truth tables	AND, OR, NOT, logic gate, truth table, Boolean algebra	Both Formative elements via LessonUp delivery system and regular summative testing. Both self and peer assessed elements via SmartRevise will be used as well.	SmartRevise self-guided revision – at least 2x20 min sessions per week on agreed focus topic Self-guided coding progression
Interactive revision 2.5	Languages Integrated Development Environment (IDE)	High and low level languages Purpose of translators Compiler vs interpreter IDE editors, error diagnostics, runtime environment and translators	Compare low vs high languages Describe purpose of translators Compare compiler vs interpreter Describe the features of an IDE	Translator, compiler, interpreter, IDE, runtime	Both Formative elements via LessonUp delivery system and regular summative testing. Both self and peer assessed elements via SmartRevise will be used as well.	SmartRevise self-guided revision – at least 2x20 min sessions per week on agreed focus topic Self-guided coding progression
Year 12	**Curriculum Coherence** **Y12 expands on many of the GCSE aspects but into much greater depth. Specifically CPUs, memory, system software, compilers, databases, networks and all elements of practical programming are taken to much higher levels of competency.**					
Medium Term Plan Title/Topic	**Themes/Concepts**	**Key Core Knowledge Foci**	**Application/Skills Foci**	**Ambitious Tier 2/3 Vocabulary**	**Assessment**	**Independent Learning**
1.1 - CPUs	Structure and function of CPU Processors Input, output and storage	Parts of CPU All registers of CPU Buses FDE CPU performance Pipelining Harvard vs VM RISC vs CISC	Explain stages of FDE cycle Describe 3 ways to improve CPU performance Describe the purpose of pipelining Compare RISC vs CISC Compare RAM / ROM	ALU, CU, PC, MAR, MDR, CIR, accumulator Buses FDE CPU performance Pipelining Harvard vs VM RISC vs CISC	Students will be tested via summative tests each fortnight (teacher marked). Self and peer-based assessment is delivered via SmartRevise continually throughout the course. EFA is delivered via live teaching	SmartRevise self-guided revision – This should be for 2 hours per week on agreed focus areas Self-guided coding progression – again this should be over 2 hours per week.

		RAM, ROM, VM, types of Secondary (inc VS)		RAM, ROM, VM		Use of RAG map to infer revision priority
1.2.1 – Systems software	Systems software Memory management Operating systems BIOS	Paging, segmentation, VM Interrupts and ISRs Scheduling methods Types of OS Device drivers Virtual machines	Explain the need for VM Explain how interrupts are used in the FDE cycle Describe types of OS Explain how the OS deals with memory management Why do we use VMs	Paging, segmentation, VM Interrupts and ISRs Scheduling – round robin, MLFQ Types of OS – distributed, embedded, real time Device drivers Virtual machines	Students will be tested via summative tests each fortnight (teacher marked). Self and peer-based assessment is delivered via SmartRevise continually throughout the course. EFA is delivered via live teaching	SmartRevise self-guided revision – This should be for 2 hours per week on agreed focus areas Self-guided coding progression – again this should be over 2 hours per week. Use of RAG map to infer revision priority
2.1.1 - Abstraction	Thinking abstractly	Nature of abstraction Need for abstraction Abstraction vs reality Devising abstract models	Describe the purpose of abstraction Explain the need for abstraction	Abstraction	Students will be tested via summative tests each fortnight (teacher marked). Self and peer-based assessment is delivered via SmartRevise continually throughout the course. EFA is delivered via live teaching	SmartRevise self-guided revision – This should be for 2 hours per week on agreed focus areas Self-guided coding progression – again this should be over 2 hours per week. Use of RAG map to infer revision priority
1.4 - Data	Data types Data Structures Boolean algebra	Primitive data types Binary/denary/hexadecimal conversions Addition of binary Floating point representation Normalisation of floating point Twos complement Signs and magnitude Masks and shifts Arrays, tuples, lists and records Linked-list, graph, stack , queue, tree and binary tree, hash tables Creating and traversing Boolean logic Karnaugh maps Simplifications (De Morgans, distribution, association, commutation, double negation) Logic gates and truth tables D type flip flops	Convert between base numbers Add and subtract binary (both normally and using twos complement) Normalise floating point numbers Explain use of various data structures Describe how you traverse linked lists and trees (depth and breadth) Complete a Karnaugh map Simplify a Boolean expression Draw logic gate diagrams and complete associated truth tables.	Primitive data types Binary/denary/hexadecimal conversions Addition of binary Floating point representation Normalisation of floating point Twos complement Signs and magnitude Masks and shifts Arrays, tuples, lists and records Linked-list, graph, stack , queue, tree and binary tree, hash tables Creating and traversing Boolean logic Karnaugh maps Simplifications (De Morgans, distribution, association, commutation, double negation) Logic gates and truth tables D type flip flops	Students will be tested via summative tests each fortnight (teacher marked). Self and peer-based assessment is delivered via SmartRevise continually throughout the course. EFA is delivered via live teaching	SmartRevise self-guided revision – This should be for 2 hours per week on agreed focus areas Self-guided coding progression – again this should be over 2 hours per week. Use of RAG map to infer revision priority
1.3 – databases and networks	Encryption and hashing Databases Networks Web technologies	Compression, encryption, hashing Relational databases, normalisation, SQL, ACID Networks TCP/IP, DNS, hardware, switching, threats, proxies, P2P and client server HTML, CSS, javascript Search engines	Explain the forms of lossless and lossy compression and their applications Draw ERDs for given scenarios Identify keys in relational DB Normalise to 3NF Write SQL statements to achieve set goals Draw and label TCP/IP stack	Compression, encryption, hashing Relational databases, normalisation, SQL, ACID Networks TCP/IP, DNS, hardware, switching, threats, proxies, P2P and client server HTML, CSS, javascript Search engines	Students will be tested via summative tests each fortnight (teacher marked). Self and peer-based assessment is delivered via SmartRevise continually throughout the course. EFA is delivered via live teaching	SmartRevise self-guided revision – This should be for 2 hours per week on agreed focus areas Self-guided coding progression – again this should be over 2 hours per week. Use of RAG map to infer revision priority

		PageRank	Create working website using HTML, CSS and Javascript Explain PageRank algorithm	PageRank		
2.1.2 -2.1.5 - Computational thinking	Thinking ahead Thinking procedurally Thinking Logically Thinking Concurrently	Inputs and outputs, Preconditions, Caching Reusable program components Identify problem and components for solution Decision points, logical design, program flow Concurrent processing – benefits and trade offs	Identify inputs and outputs for given scenario. Describe preconditions. Describe the benefits of caching Compare concurrency to parallel programming elements	Preconditions, caching, concurrent	Students will be tested via summative tests each fortnight (teacher marked). Self and peer-based assessment is delivered via SmartRevise continually throughout the course. EFA is delivered via live teaching	SmartRevise self-guided revision – This should be for 2 hours per week on agreed focus areas Self-guided coding progression – again this should be over 2 hours per week. Use of RAG map to infer revision priority
2.3.1 - ALgorithms	Algorithms, Big O Notation Complexity in algorithms Data structure algorithms Searching and sorting algorithms	Big O notation: Constant, linear, polynomial, exponential, logarithmic Algorithms for Stacks, queues, trees, linked lists Bubble, insertion, merge and quick sort A* and Dijkstra Linear and binary search	Compare space complexity and time complexity of specific searches and sorts Perform A* or Dijkstra's algorithm on provided graph. Describe steps in binary search	Big O notation: Constant, linear, polynomial, exponential, logarithmic Algorithms for Stacks, queues, trees, linked lists Bubble, insertion, merge and quick sort A* and Dijkstra Linear and binary search	Students will be tested via summative tests each fortnight (teacher marked). Self and peer-based assessment is delivered via SmartRevise continually throughout the course. EFA is delivered via live teaching	SmartRevise self-guided revision – This should be for 2 hours per week on agreed focus areas Self-guided coding progression – again this should be over 2 hours per week. Use of RAG map to infer revision priority
Project Analysis	Project overview Research Analysis Success criteria Hardware/software reqs	Propose project with key elements Conduct research and interviews Perform analysis Create list of requirements (success criteria) List requirements for hw/sw	Compare similar products and identify similarities and differences. Justify interview questions Justify all success criteria linking back to research gathering	Requirement specification Analysis Success criteria Stakeholders	Students will be tested via summative tests each fortnight (teacher marked). Self and peer-based assessment is delivered via SmartRevise continually throughout the course. EFA is delivered via live teaching	SmartRevise self-guided revision – This should be for 2 hours per week on agreed focus areas Self-guided coding progression – again this should be over 2 hours per week. Use of RAG map to infer revision priority
Project Design	Design Specification Test Plan	Structure diagram Variable declarations Data structures Class definitions Flowcharts Pseudocode Screen mock ups Validation routines	Break down system into component parts Justify variables Justify data structures Plan, create accurate flow diagrams Implement suitable validation into designs	Structure diagram Variable declarations Data structures Class definitions Flowcharts Pseudocode Screen mock ups Validation routines	Students will be tested via summative tests each fortnight (teacher marked). Self and peer-based assessment is delivered via SmartRevise continually throughout the course. EFA is delivered via live teaching	SmartRevise self-guided revision – This should be for 2 hours per week on agreed focus areas Self-guided coding progression – again this should be over 2 hours per week. Use of RAG map to infer revision priority
Year 13	**Curriculum Coherence**					
	Y13 is almost evenly split between time dedicated to finishing theory, revising the theory and completing the coursework project. The project aims to be completed by Easter to give ample time to recap and refresh topics delivered early in Y12.					
Medium Term Plan Title/Topic	**Themes/Concepts**	**Key Core Knowledge Foci**	**Application/Skills Foci**	**Ambitious Tier 2/3 Vocabulary**	**Assessment**	**Independent Learning**

1.2.2-1.2.3	Application generation Software Development	Applications, Utilities, Sources, Translators, Compilation stages, Linkers, Loaders, Libraries Software development methodologies	Describe how a compiler turns a high level language into machine code. Describe the role of linkers, loaders and libraries when compiling code. Compare the software development methodologies	Applications, Utilities, Sources, Translators, Compilation stages, Linkers, Loaders, Libraries Software development methodologies	Students will be tested via summative tests each fortnight (teacher marked). Self and peer-based assessment is delivered via SmartRevise continually throughout the course. EFA is delivered via live teaching	SmartRevise self-guided revision – This should be for 2 hours per week on agreed focus areas Self-guided coding progression – again this should be over 2 hours per week. Use of RAG map to infer revision priority
Project A/B finalise	Project overview Research Analysis Success criteria Hardware/software reqs Design Specification Test Plan	Propose project with key elements Conduct research and interviews Perform analysis Create list of requirements (success criteria) List requirements for hw/sw Structure diagram Variable declarations Data structures Class definitions Flowcharts Pseudocode Screen mock ups Validation routines	Compare similar products and identify similarities and differences. Justify interview questions Justify all success criteria linking back to research gathering Break down system into component parts Justify variables Justify data structures Plan, create accurate flow diagrams Implement suitable validation into designs	Requirement specification Analysis Success criteria Stakeholders Structure diagram Variable declarations Data structures Class definitions Flowcharts Pseudocode Screen mock ups Validation routines	Students will be tested via summative tests each fortnight (teacher marked). Self and peer-based assessment is delivered via SmartRevise continually throughout the course. EFA is delivered via live teaching	SmartRevise self-guided revision – This should be for 2 hours per week on agreed focus areas Self-guided coding progression – again this should be over 2 hours per week. Use of RAG map to infer revision priority
2.2.1 – 2.2.2	Programming Techniques Computational Methods	Constructs Recursion Global and local variables Functions and procedures IDEs Problem recognition and solving Divide and conquer Backtracking, data mining, heuristics, performance modelling, pipelining, visualisation	Create modular programs Describe the use of recursion Explain the benefits of data mining Describe why we use heuristics	Constructs Recursion Global and local variables Functions and procedures IDEs Problem recognition and solving Divide and conquer Backtracking, data mining, heuristics, performance modelling, pipelining, visualisation	Students will be tested via summative tests each fortnight (teacher marked). Self and peer-based assessment is delivered via SmartRevise continually throughout the course. EFA is delivered via live teaching	SmartRevise self-guided revision – This should be for 2 hours per week on agreed focus areas Self-guided coding progression – again this should be over 2 hours per week. Use of RAG map to infer revision priority
1.2.4	Types of programming languages Procedural OOL Addressing memory	LMC Immediate, direct, indirect, indexed Classes, objects, methods, attributes, inheritance, encapsulation, polymorphism	Explain the use of OOL to create code (merits and drawbacks) Apply addressing techniques to set problems. Write LMC code to solve set problems	LMC Immediate, direct, indirect, indexed Classes, objects, methods, attributes, inheritance, encapsulation, polymorphism	Students will be tested via summative tests each fortnight (teacher marked). Self and peer-based assessment is delivered via SmartRevise continually throughout the course. EFA is delivered via live teaching	SmartRevise self-guided revision – This should be for 2 hours per week on agreed focus areas Self-guided coding progression – again this should be over 2 hours per week. Use of RAG map to infer revision priority
Project development	Code development and documentation	Document full development journey Intent, coding progression, testing and remedial actions	Successfully code planned software against design documentation Test and conduct debugging fixes and retest	Implementation, debugging, remedial action.	Students will be tested via summative tests each fortnight (teacher marked). Self and peer-based assessment is delivered via SmartRevise	SmartRevise self-guided revision – This should be for 2 hours per week on agreed focus areas

					continually throughout the course. EFA is delivered via live teaching	Self-guided coding progression – again this should be over 2 hours per week. Use of RAG map to infer revision priority
Project Evaluation	Testing Evaluation	Terminal testing End user testing Reference to initial success criteria	Explain strengths and weaknesses of project cycle. Detail how missing or non functional elements could be addressed. Describe potential extensions to the project code.	Terminal testing End user testing	Students will be tested via summative tests each fortnight (teacher marked). Self and peer-based assessment is delivered via SmartRevise continually throughout the course. EFA is delivered via live teaching	SmartRevise self-guided revision – This should be all 4 hours of study time outside of the classroom at this stage Use of RAG map to infer revision priority
Revision routines and recaps	All covered topics	All content	All content	All content	Students will be tested via summative tests each fortnight (teacher marked). Self and peer-based assessment is delivered via SmartRevise continually throughout the course. EFA is delivered via live teaching	SmartRevise self-guided revision – This should be all 4 hours of study time outside of the classroom at this stage Use of RAG map to infer revision priority