

Year 7 9 week rotation

Year	Term	Unit #	Unit Title	Curriculum Coverage
7	Aut 1	1	Online Safety + Systems Architecture (<i>Weeks</i>	Week 1: Online Safety (e.g., cyberbullying, privacy settings, safe social media
				Weeks 2–5: CPU basics (CU, ALU), memory vs. storage, hardware/software
7	Aut 2	2	Data Representation (<i>Weeks 6–9</i>)	Binary & decimal, basic conversions, character encoding (ASCII)

Year 8

Year	Term	Unit #	Unit Title	Curriculum Coverage
8	Aut 1	7	Computer Systems	Hardware vs. software, CPU basics, memory, storage devices
8	Aut 2	8	Advanced Data Representation	Binary/hex conversions, ASCII/Unicode, optional compression
8	Spr 1	9	Computer Networks	LAN vs. WAN, topologies, wired vs. wireless, router/switch
8	Spr 2	10	Network Threats	Malware (virus, trojan, spyware), social engineering, firewall/encryption basics
8	Sum 1	11	Algorithms (Pseudo/Flowcharts)	Input/process/output, decisions, iteration (loops), step-by-step solutions
8	Sum 2	12	Python Programming	Variables, data types, selection & iteration, basic coding tasks

Year 9

Year	Term	Unit #	Unit Title	Curriculum Coverage
9	Aut 1 (4	13	Foundations of Computing (4-wk rotation)	E-safety & responsible use, basic hardware/software, quick Python intro
9	Aut 2 & 3	14	Systems Arch & Memory, Data Rep (part of 17-wk block)	CPU (CU, ALU), FDE cycle, RAM vs. ROM, secondary storage, deeper binary/hex
9	Aut 2 & 3	15	Boolean Logic & Early Algorithms, Intro Programming	AND, OR, NOT gates, truth tables, simple searching/sorting, basic coding (IF/LOC)
9	Spr 2 & 3	16	Networks & Protocols (2nd 17-wk block)	LAN, WAN, topologies, TCP/IP model, wired vs. wireless; how data is transmitted
9	Spr 2 & 3	17	Network Security & Deeper Algorithms	Malware, social engineering, advanced security measures; expansions on searching
9	Sum 2	18	Consolidation & Project	Capstone combining networks, security, programming skills

Year 10

Year	Term	Unit #	Unit Title	Curriculum Coverage
------	------	--------	------------	---------------------

10	Aut 1	1.1	CPU & Systems Architecture (+1 wk Python)	CPU components (CU, ALU, registers), embedded systems, FDE cycle
10	Aut 2	1.2	Memory & Storage (+1 wk Python)	RAM vs. ROM, secondary storage types, capacity calculations
10	Spr 1	1.3	Networks & Protocols (+1 wk Python)	LAN vs. WAN, topologies, protocols (TCP/IP), client-server vs. peer-to-peer
10	Spr 2	1.4	Network Security (+1 wk Python)	Social engineering, malware, firewalls, encryption, prevention methods
10	Sum 1	1.5	Systems Software (+1 wk Python)	Operating systems (functions, memory mgmt, file mgmt), utility software
10	Sum 2	1.6	Ethical, Legal, Cultural, Environmental (+1 wk P	Data protection, open source vs. proprietary, environmental impact, legal issues

Year 11

Year	Term	Unit #	Unit Title	Curriculum Coverage
11	Aut 1	Paper 2	Algorithms & Prog Fundamentals (2.1 & 2.2)	Searching/sorting, pseudocode, selection, iteration; code building blocks
11	Aut 2	Paper 2	Robust Programs & Boolean Logic (2.3 & 2.4)	Input validation, testing, logic gates (AND/OR/NOT), truth tables
11	Spr 1	Paper 2	Prog Languages & IDEs (2.5) + Recap (Paper 2)	High- vs. low-level languages, compilers/interpreters, debugging; plus overall Pa
11	Spr 2	Revision	Paper 2 (and minimal Paper 1 if needed)	Consolidation of all topics; targeted exam prep
11	Sum 1 & Exams		Final GCSE Exams	N/A

Sequencing: Knowledge & Skills	Literacy & Reading	Differentiation
- Reinforce e-safety principles	KW: e-safety, CPU, ALU, RAM, ROM, hardware,	MA: Explore fetch–decode–execute cycle;
- Intro to how a computer processes data (input–process–		LA: focus on basic hardware I/O & safe
Build numeracy with place values & bits/bytes	KW: binary, decimal, ASCII, bit, byte	MA: Binary arithmetic;
		LA: guided conversions

Sequencing: Knowledge & Skills	Literacy & Reading	Differentiation
Establish fundamental concepts of how a computer is built	KW: CPU, RAM, ROM, hardware, software	MA: Detailed CPU registers; LA: focus on input/output roles
Deepen representation knowledge with practical conversions	KW: binary, hexadecimal, ASCII, Unicode	MA: Advanced hex arithmetic; LA: step-by-step conversions
Explore how data travels; compare network setups	KW: network, topology, LAN, WAN, router, switch	MA: Analyze real network diagrams; LA: simpler topologies
Understand threats & basic cyber defenses	KW: malware, virus, phishing, firewall, encryption	MA: Deeper infiltration testing; LA: definitions of threats
Develop logical thinking & methodical problem-solving	KW: algorithm, flowchart, pseudocode, decision	MA: More complex branching; LA: step-based flowcharts
Apply programming constructs to small solutions	KW: variable, iteration, selection, syntax	MA: Small projects; LA: structured tasks w/ partial code

Sequencing: Knowledge & Skills	Literacy & Reading	Differentiation
Reestablish computing basics & digital citizenship	KW: e-safety, hardware, software, Python syntax	MA: small coding tasks; LA: simpler demos
From CPU fundamentals to memory & data representation	KW: CPU, FDE, registers, binary, hex	MA: deeper FDE details; LA: bullet-point approach
Extend logic to practical algorithms & code	KW: boolean, flowcharts, searching/sorting, pseudo	MA: implement bubble sort; LA: outline logic w/ partial code
Networking theory; from physical to protocols	KW: LAN, WAN, IP, packet, protocol	MA: deeper packet switching; LA: simpler diagrams
Threat awareness + advanced coding practice	KW: malware, phishing, hashing, iteration	MA: infiltration scenarios; LA: step-by-step tasks
Final integration & demonstration of knowledge	KW: project scope, documentation	MA: lead project design; LA: guided instructions

Sequencing: Knowledge & Skills	Literacy & Reading	Differentiation
--------------------------------	--------------------	-----------------

Intro to how instructions are processed within the CPU	KW: CPU, ALU, registers, FDE, embedded systems	MA: multiple cores; LA: guided FDE cycle
Deeper look at how data is stored/retrieved	KW: RAM, ROM, SSD, HDD, optical, capacity	MA: complex comparisons; LA: simple pros/cons
Building from hardware to fundamental networking	KW: LAN, WAN, IP, MAC, DNS	MA: deeper packet switching; LA: simpler diagrams
Extending networking into security aspects	KW: malware, firewall, encryption, phishing	MA: advanced hacking scenarios; LA: basic threat definitions
Deeper OS understanding & tasks	KW: OS, utility, encryption, defrag, backup	MA: advanced OS case studies; LA: OS basics
Reflect on societal & ethical factors	KW: ethics, law, environment, open-source vs. propri	MA: real laws/case studies; LA: scenario tasks

Sequencing: Knowledge & Skills

Core algorithmic thinking => code building

Building reliable code with thorough testing & logic

Understand how code is written & translated

Recap 2.1–2.5 + key Paper 1 points

N/A

Literacy & Reading

KW: linear/binary search, bubble sort, iteration, selection

KW: validation, test plans, logic gates, truth tables

KW: compiler, interpreter, debugging, syntax errors

N/A: referencing existing keywords

N/A

Differentiation

MA: implement code fully;

LA: flowcharts only

MA: create & test robust code;

LA: step-based debugging

MA: examine assembly;

LA: focus on high-level usage

Focus on bridging knowledge gaps

N/A

Enrichment & Cultural Refs	British Values & SMSC	Personal Dev
E: Real-world examples (smartphones, consoles); Safe/responsible tech	SMSC: Importance of respectful, safe behavior	Safe & correct use of devices; digital
E: Digital images & text (ASCII art)	SMSC: Historical context of computing	Develop logical thinking

Enrichment & Cultural Refs	British Values & SMSC	Personal Dev
E: Evolution (mainframes → smartphones)	Responsible use of tech & safe practice	Aware of how computers process data & cor
E: How data representation affects images, text, audio	Historical context (WWII codebreaking)	Numeracy & problem-solving
E: Real-world usage (home or school networks)	Collaboration & standardization across networks	Communication & teamwork
E: Cyber-attack news stories	Ethical approach to cybersecurity & privacy	Personal data security awareness
E: Daily algorithms (search engines, GPS)	Intellectual curiosity & problem-solving resilience	Analytical thinking
E: Coding clubs, competitions (cybersecurity)	Fostering creativity & discipline in coding	Resilience in debugging

Enrichment & Cultural Refs	British Values & SMSC	Personal Dev
E: Real-life hardware usage	Tolerance & respect for tech usage	Broad digital literacy foundation
E: History of CPU design, big tech data usage	Impact on society, memory footprints	Improve hardware & data literacy
E: Turing & famous algorithms	Logical, ethical problem-solving	Computational thinking
E: Evolution of the internet, real use cases	Respecting global infrastructure & laws	Understand global connectivity
E: High-profile hacking stories	Cybersecurity ethics & data protection	Secure mindsets with tech use
E: Combining hardware + software + security	Respect for intellectual property	Ownership of final project

Enrichment & Cultural Refs	British Values & SMSC	Personal Dev
----------------------------	-----------------------	--------------

E: Von Neumann architecture

Respect tech's power

Interest in comp engineering

E: Big data, cloud in daily life

Responsible data storage

Hardware constraints understanding

E: Real usage in home/school networks

Infrastructure connecting us

Recognize safe vs. unsafe networks

E: Real data-breach news

Ethical & legal hacking aspects

Adopting secure online habits

E: Linux vs. Windows, open-source communities

Daily computing knowledge

Problem-solving with system software

E: E-waste, green computing initiatives

Balance tech & environment

Ethical & moral tech use

Enrichment & Cultural Refs

British Values & SMSC

Personal Dev

E: Historical sorting (Knuth)

Problem-solving curiosity

Strong logic skills

E: Real software failures

Responsibility in coding

Learning from mistakes (testing)

E: Open-source vs. proprietary dev communities

Freedom to innovate w/ licensing

Confidence in coding & debugging

E: Peer study, group revision

Collaboration & mutual respect

Exam resilience & time management

N/A

N/A

N/A

Careers	Assessment
Hardware engineer, tech	End of topic test
Data analytics, software dev	End of rotation

Careers	Assessment
IT technician, hardware engine	End of topic test
Data analyst, programmer	Interim assessment
Network engineer	End of topic test
Cybersecurity analyst	Interim assessment
Software developer	End of topic test
Python dev, AI specialist	Interim assessment

Careers	Assessment
Various computing roles	Short tasks / checkpoint
Engineer, hardware design	End-of-block test
Programmer, data scientist	Mid-block assessment
Network engineer	End-of-block test
Cybersecurity specialist	Mid-block tests
Developer, project manager	End-of-year summative

Careers	Assessment
---------	------------

Hardware engineer	End-of-unit test
Systems admin	End-of-unit test
Network engineer	Mock or end-of-unit
Cybersecurity specialist	End-of-unit test
Software engineer	End-of-unit test
IT lawyer, policy advisor	End-of-year exam

Careers

Assessment

Software developer	Mock exam (Paper 2)
QA engineer, tester	Mock exam or practice papers
DevOps, sys analyst	Mock exam
Any computing path	Full practice papers
N/A	Public Exams