GCE ADVANCED LEVEL · INFORMATION & COMMUNICATION TECHNOLOGY

A-Level ICT Course Syllabus

A unified, non-overlapping 7-module curriculum arranged chronologically from Lower Sixth foundations to Upper Sixth mastery, structured into Modules, Chapters, and Lessons.

SOURCE: MINESEC NATIONAL HARMONISED PROGRESSION SHEETS (LOWER SIXTH & UPPER SIXTH) · ORIGIN TAGS SHOW PROGRESSION (L6 → U6)

198

7
SELF-CONTAINED MODULES

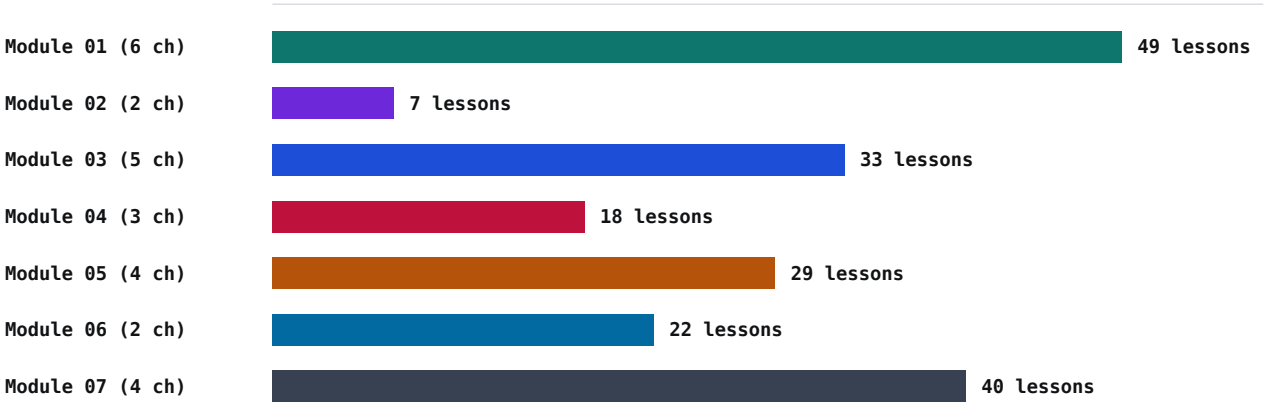
26
CHAPTERS

0
CROSS-MODULE SPLITS

Every ICT domain lives inside a single self-contained module with zero cross-class splitting, while lessons inside each chapter progress chronologically from Lower Sixth (L6) through Upper Sixth (U6).

CURRICULUM BREAKDOWN

Lessons by Thematic Module



Module 01 — Computing Systems, Hardware, OS & Emerging Technologies (6 Chapters · 49 Lessons)

Module 02 — Data Representation & Digital Logic (2 Chapters · 7 Lessons)

Module 03 — Networking, Data Communication & Internet Technologies (5 Chapters · 33 Lessons)

Module 04 — Cybersecurity, Digital Citizenship, Ethics & Green Computing (3 Chapters · 18 Lessons)

Module 05 — Information Systems, SDLC, Databases & IT Project Management (4 Chapters · 29 Lessons)

Module 06 — Productivity Tools, Multimedia & Web Development (2 Chapters · 22 Lessons)

Module 07 — Data Structures, Algorithms, Software Design & Programming (4 Chapters · 40 Lessons)

MODULE 01 · 6 CHAPTERS · 49 LESSONS

Computing Systems, Hardware, OS & Emerging Technologies

Unifies computer history, hardware architecture, primary/secondary storage, operating systems, maintenance, ergonomics, assistive technology, AI/Machine Learning, robotics, VR/AR, IoT, and cloud computing into one self-contained chronological module.

CHAPTER 1.1 · 10 LESSONS (CHRONOLOGICAL L6 → U6)

History, Classification & Architecture of Computers

Competency: Compare computer generations, architectures (Von Neumann/Harvard, CISC/RISC, Flynn), storage, and peripherals.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
1.1.1 L6 · L1	**History and Evolution of Computing**	Identify computer generations and technologies; compare size, power, and cost; differentiate Von Neumann vs. Harvard architecture and stored-program concept.
1.1.2 L6 · L7	**Types of Computers**	Identify supercomputers, mainframes, minicomputers, and microcomputers; compare size, power, cost, and purpose.
1.1.3 L6 · L8	**Basic Components of a Computer**	Explain hardware, input, output, storage, and processing devices; choose appropriate devices for a given situation.
1.1.4 L6 · L9	**Input Devices**	Explain automatic data capture (MICR, OCR, OMR, barcode, QR, card, RFID, biometric); explain how AI image recognition extends data capture.
1.1.5 L6 · L10	**Output Devices**	Differentiate dot-matrix, laser, and inkjet printers; describe projectors, plotters, 3D printers, and actuators; justify device choices.
1.1.6 L6 · L11	**Secondary Storage Media and Devices**	Differentiate secondary vs. primary storage; describe magnetic, optical, and solid-state storage; justify storage choices.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
1.1.7 L6 · L12	**Primary Storage Devices**	Describe RAM, ROM, cache, and registers; compare speed, cost, and capacity; explain why AI/ML training needs large dataset storage.
1.1.8 L6 · L13	**Processing Device & Machine Instruction Cycle**	Differentiate CPU and GPU; describe stages of the machine instruction cycle; explain why AI/ML relies on parallel GPU processing.
1.1.9 L6 · L14	**Processor Architectures, Parallel & Distributed Computing**	Explain CISC vs. RISC; describe Flynn's taxonomy (SISD, SIMD, MISD, MIMD); differentiate parallel and distributed computing.
1.1.10 L6 · L15	**Conversion Between Storage & Processing Units**	Identify storage (bits–TB) and processing (Hz–GHz) units; convert between units of storage.

CHAPTER 1.2 · 9 LESSONS (CHRONOLOGICAL L6 → U6)

Computer Software & Operating Systems

Competency: Classify system/application software, explain OS scheduling and memory/file/I/O management, and use GUI/CLI.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
1.2.1 L6 · L16	**Definitions and Classification of Software**	Explain software; differentiate system vs. application software and open-source vs. proprietary software.
1.2.2 L6 · L17	**Application Software**	Differentiate general-purpose, specific-purpose, and tailor-made software; describe common categories and choose software for tasks.
1.2.3 L6 · L18	**System Software and Examples**	Describe OS, device drivers, and utilities (antivirus, disk cleaner, defragmenter, compression); identify AI-powered utilities.
1.2.4 L6 · L19	**Notions of the Operating System**	Outline OS history; describe OS types (single-user, batch, online, multi-user, NOS, multitasking, real-time); state OS functions.
1.2.5 L6 · L20	**Functions of an Operating System 1 (Scheduling)**	Distinguish pre-emptive vs. non-pre-emptive scheduling; describe FCFS, SJF, SRT, Round Robin; explain AI predictive scheduling.
1.2.6 L6 · L21	**Functions of an Operating System 2 (Memory, Files, I/O)**	Explain memory, file, and device management; define deallocation, virtual memory, buffering, spooling, and metadata.
1.2.7 L6 · L22	**Installing an Operating System & User Interfaces**	Install an OS (Windows/Linux); compare GUI and CLI strengths/weaknesses; select appropriate interface.
1.2.8 L6 · L23	**Using the GUI of an Operating System**	Explain file, folder, and file format; perform file/folder operations and set up OS access protection via GUI.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
1.2.9 L6 · L24	**Using the CLI of an Operating System**	Launch MS-DOS/Linux CLI; perform file/folder operations via CLI; write and execute a CLI automation script.

CHAPTER 1.3 · 5 LESSONS (CHRONOLOGICAL L6 → U6)

Maintenance, Ergonomics & Assistive Technology

Competency: Identify hardware/software faults, apply ergonomic setup, and configure assistive technologies.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
1.3.1 L6 · L25	**Hardware Faults Identification and Correction**	Explain computer, preventive, and corrective maintenance; describe common hardware faults and fixes.
1.3.2 L6 · L26	**Software Faults Identification and Correction**	Differentiate hardware vs. software maintenance; describe common software faults and fixes.
1.3.3 L6 · L27	**Assistive Technology**	Explain assistive technology (Braille keyboard, audio, ASR); explain how AI speech recognition and predictive text improve accessibility.
1.3.4 L6 · L28	**Computer Ergonomics**	Explain ergonomics and computer-related health hazards (MSDs, eye strain); describe correct posture, habits, and setup.
1.3.5 L6 · IA1	**Integration Activity No 1 (System Setup & Accessibility)**	Select and justify system/application software, install/configure OS with ergonomic setup, and demonstrate an accessibility feature.

CHAPTER 1.4 · 15 LESSONS (CHRONOLOGICAL L6 → U6)

Artificial Intelligence, Machine Learning & Intelligent Systems

Competency: Explain AI/ML/DL/LLMs, ethical AI use, Teachable Machine/visual ML, and AI system development.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
1.4.1 L6 · L2	**Introduction to Artificial Intelligence (AI)**	Define AI and explain its history/evolution; identify main types of AI; explain common AI applications.
1.4.2 L6 · L3	**AI Ethics and Responsible Use**	Explain ethical issues in AI; examine risks and benefits; discuss principles for responsible AI use.
1.4.3 L6 · L4	**AI Techniques and Intelligent Systems**	Describe common AI techniques and applications; explain characteristics of intelligent systems and compare them.
1.4.4 L6 · L5	**Machine Learning**	Define Machine Learning; explain basic ML principles; distinguish supervised, unsupervised, and reinforcement learning.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
1.4.5 L6 · L6	**Developing AI Systems**	Explain stages in developing an AI system; identify AI languages/tools; explain why Python is widely used.
1.4.6 L6 · L55	**Multimodal Systems**	Differentiate unimodal vs. multimodal systems; explain how multimodal systems work with real-world examples.
1.4.7 L6 · L56	**AI Technologies (ML, Neural Networks & Deep Learning)**	Differentiate ML, neural networks, and deep learning; match domains to techniques; train a simple visual ML model.
1.4.8 L6 · L57	**Generative AI**	Explain generative AI and LLMs; explain hallucinations and bias sources; evaluate impact and discuss ethical use.
1.4.9 L6 · L58	**Using Generative AI for Day-to-Day Activities**	Formulate structured prompts for workplace/daily tasks; refine prompts and write effective follow-up prompts.
1.4.10 U6 · L1	**The Concept of Artificial Intelligence**	Define AI and explain its history/development; identify main types of AI and common application areas.
1.4.11 U6 · L2	**AI Ethics and Responsible Use**	Explain ethical issues in AI; examine risks and benefits; discuss principles for responsible AI use.
1.4.12 U6 · L3	**AI Techniques and Intelligent Systems**	Describe AI techniques; explain characteristics of intelligent systems (smart school, city, health, home) and compare them.
1.4.13 U6 · L4	**Machine Learning (Teachable Machine)**	Define ML and its relationship with AI; explain ML principles; build and test simple ML models using Teachable Machine.
1.4.14 U6 · L5	**Developing AI Systems**	Describe major stages of developing an AI system; identify AI languages/tools; explain why Python is widely used.
1.4.15 U6 · L6	**Integration Activities (AI Systems)**	Consolidate AI and Machine Learning concepts through applied exercises.

CHAPTER 1.5 · 6 LESSONS (CHRONOLOGICAL L6 → U6)

Robotics, Control Systems, Simulation, VR/AR & IoT

Competency: Explain industrial/scientific automation, robotics, monitoring/control, simulation, embedded/IoT, and VR/AR.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
1.5.1 L6 · L48	**Computer Systems in Industry and Science**	Describe applications in weather forecasting, CAD, CAM, image processing, and simulation; evaluate industrial automation.
1.5.2 L6 · L49	**Robots and Their Application in Daily Life**	Describe a robot; explain applications in manufacturing, health, and home with examples.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
1.5.3 L6 · L50	**Monitoring, Control, and Monitoring & Control Systems**	Differentiate monitoring, control, and combined systems; explain how each works with real-life applications.
1.5.4 L6 · L51	**Simulation and Modelling**	Differentiate simulation and modelling; explain pros/cons; describe applications and design a technology simulation.
1.5.5 L6 · L52	**Embedded Systems and IoT**	Differentiate embedded systems and IoT; explain how both work and describe real-life applications.
1.5.6 L6 · L53	**Virtual Reality and Augmented Reality**	Explain VR and AR; describe their roles across domains; differentiate AR and VR systems.

CHAPTER 1.6 · 4 LESSONS (CHRONOLOGICAL L6 → U6)

Cloud Computing

Competency: Compare SaaS/PaaS/IaaS, Public/Private/Hybrid deployment models, shared responsibility, and cloud ethics.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
1.6.1 U6 · L73	**Introduction to Cloud Computing**	Explain cloud vs. local computing, core components (data centre, virtualisation), consumer tools, and pros/cons.
1.6.2 U6 · L74	**Types of Cloud Computing Services**	Compare SaaS, PaaS, and IaaS; classify common digital platforms into SaaS, PaaS, and IaaS.
1.6.3 U6 · L75	**Deployment Models, Security & Cloud Ethics**	Differentiate Public, Private, Hybrid clouds; evaluate privacy trade-offs, shared responsibility, outages, and vendor lock-in.
1.6.4 U6 · L76	**Integration Activity (Cloud Computing)**	Cloud architecture and ethics debate.

MODULE 02 · 2 CHAPTERS · 7 LESSONS

Data Representation & Digital Logic

Complete coverage of number systems, negative number representations (two's complement), base arithmetic, logic gates, logic circuits, De Morgan's laws, and Boolean expression simplification.

CHAPTER 2.1 · 3 LESSONS (CHRONOLOGICAL L6 → U6)

Number Systems, Negative Numbers & Base Arithmetic

Competency: Convert across bases 2, 8, 10, 16, represent negative numbers, and perform base

arithmetic.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
2.1.1 L6 · L71	**Introduction to Data Representation & Number Systems**	Identify binary, octal, decimal, hex; convert base 10 to base 2, 8, 16; differentiate bit, nibble, byte, word.
2.1.2 L6 · L72	**Conversion Between Number Systems & Negative Numbers**	Convert base 2, 8, 16 to base 10; describe sign-magnitude, one's complement, and two's complement; compute complements.
2.1.3 L6 · L73	**Subtraction and Representing Negative Numbers**	Perform addition and subtraction in base 2, 8, and 16.

CHAPTER 2.2 · 4 LESSONS (CHRONOLOGICAL L6 → U6)

Logic Gates, Circuits, De Morgan's Laws & Boolean Simplification

Competency: Draw truth tables, convert between circuits and Boolean expressions, and simplify using De Morgan's laws.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
2.2.1 L6 · L74	**Logic Gates**	Identify OR, AND, NOT, NAND, NOR, XOR, XNOR gates; sketch symbols; draw truth tables (up to 3 inputs).
2.2.2 L6 · L75	**Logic Circuits**	Explain logic circuits and Boolean expressions; derive circuits from expressions and expressions from circuits; build truth tables.
2.2.3 L6 · L76	**Boolean Expressions from Truth Tables & De Morgan's Laws**	Derive Boolean expressions from truth tables; state and prove De Morgan's laws.
2.2.4 L6 · L77	**Simplification of Boolean Expressions**	Explain benefits of simplification; state Boolean laws; simplify expressions using Boolean laws and De Morgan's laws.

MODULE 03 · 5 CHAPTERS · 33 LESSONS

Networking, Data Communication & Internet Technologies

All networking fundamentals, coverage classifications, architectures, topologies, equipment, transmission media/modes, error detection, TCP/IP & OSI model, internet services, teleworking, e-services, and e-learning.

CHAPTER 3.1 · 6 LESSONS (CHRONOLOGICAL L6 → U6)

Fundamentals & Classification of Computer Networks

Competency: Compare networked vs. standalone systems, classify PAN/LAN/CAN/MAN/WAN, and select

organisational networks.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
3.1.1 U6 · L7	**Fundamentals of Computer Networks**	Describe computer networks; compare networked vs. standalone systems; identify real-world examples and benefits/limitations.
3.1.2 U6 · L8	**Network Complexity and Operational Challenges**	Explain complexity in setting up/maintaining networks and interoperability between heterogeneous systems.
3.1.3 U6 · L9	**Integration Activities (Network Fundamentals)**	Review network fundamentals and operational challenges.
3.1.4 U6 · L10	**Types of Networks by Coverage Area**	Describe and classify LAN, MAN, WAN, CAN, PAN; compare coverage, speed, cost, and device/user capacity.
3.1.5 U6 · L11	**Choosing the Right Network for an Organization**	Identify organisational networking needs; select and evaluate the most appropriate network type.
3.1.6 U6 · L12	**Integration Activities (Network Classification)**	Organisational network classification scenarios.

CHAPTER 3.2 · 8 LESSONS (CHRONOLOGICAL L6 → U6)

Network Architectures, Topologies & Equipment

Competency: Compare peer-to-peer vs. client/server, physical topologies, and configure network equipment.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
3.2.1 U6 · L13	**Network Architecture (Peer-to-Peer & Client/Server)**	Explain logical node relationships; compare advantages and disadvantages; choose architecture for an organisation.
3.2.2 U6 · L14	**Types of Physical Network Topologies**	Describe mesh, star, ring, bus, and tree topologies; identify physical layouts and pros/cons.
3.2.3 U6 · L15	**Comparing and Choosing Network Topologies**	Compare topologies; evaluate and propose a suitable topology for specific organisational needs.
3.2.4 U6 · L16	**Integration Activities (Topologies & Architecture)**	Network architecture and topology design exercises.
3.2.5 U6 · L17	**Network Equipment**	Explain and differentiate roles of modem, hub, router, NOS, etc.; identify required network components.
3.2.6 U6 · L18	**Setting Up a Computer Network**	Demonstrate steps to set up a basic network; interconnect essential equipment; set up and test a peer-to-peer network.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
3.2.7 U6 · L19	**Selecting Equipment for Organizational Networks**	Select appropriate equipment for an organisation (e.g. a school); explain selection factors and plan setup.
3.2.8 U6 · L20	**Integration Activities (Network Setup)**	Practical network equipment selection and setup.

CHAPTER 3.3 · 6 LESSONS (CHRONOLOGICAL L6 → U6)

Data Transmission Modes, Media, Device Control & Error Detection

Competency: Differentiate transmission modes/types, guided/unguided media, peripheral control, and parity/error checks.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
3.3.1 U6 · L27	**Modes of Data Transmission**	Explain logical topology methods (CSMA/CD, CSMA/CA); describe simplex, half-duplex, full-duplex and illustrate signal flow.
3.3.2 U6 · L28	**Broadband and Narrowband Transmission**	Define and differentiate broadband vs. narrowband transmission; identify appropriate transmission types.
3.3.3 U6 · L29	**Types of Data Transmission**	Contrast serial vs. parallel and synchronous vs. asynchronous transmission; illustrate analog vs. digital signals.
3.3.4 U6 · L30	**Transmission Media (Guided and Unguided)**	Describe guided (copper, coax, fibre) and unguided (Bluetooth, microwave, Wi-Fi) media; choose medium by context.
3.3.5 U6 · L31	**Peripheral Device Control Mechanisms**	Explain buffers, interrupts, interrupt priorities, polling, and handshaking; determine control technique in a scenario.
3.3.6 U6 · L35	**Error Detection and Data Transfer Checks**	Identify transmission errors; describe data transfer checking mechanisms and explain parity checks.

CHAPTER 3.4 · 3 LESSONS (CHRONOLOGICAL L6 → U6)

Network Protocols & the OSI Reference Model

Competency: Explain TCP/IP, HTTP, and map protocols to the 7 layers of the OSI reference model.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
3.4.1 U6 · L32	**Network Protocols and Their Roles**	Explain meaning/role of protocols; discuss IP, TCP, HTTP; evaluate how TCP/IP enables reliable transmission.
3.4.2 U6 · L33	**Protocol Suites and the OSI Reference Model**	Explain protocol suites; describe the 7 layers of the OSI model and map network protocols to OSI layers.
3.4.3 U6 · L34	**Integration Activities (Transmission & Protocols)**	Data transmission and OSI model review.

CHAPTER 3.5 · 10 LESSONS (CHRONOLOGICAL L6 → U6)

Internet, Social Media, Electronic Services & E-Learning

Competency: Use internet/retrieval services, teleworking, e-commerce/e-health/e-government, and CAL/e-learning.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
3.5.1 U6 · L36	History and Development of the Internet	Define the Internet and its purpose; recall major historical events, growth factors, and pros/cons.
3.5.2 U6 · L37	Internet Connectivity Technologies & Requirements	Identify Internet connection technologies; describe roles of connectivity devices; use core browser features.
3.5.3 U6 · L38	Social Networks and Collaborative Environments	Identify social media/collaborative platforms; describe impacts, propose responsible use, and demonstrate teamwork.
3.5.4 U6 · L39	Internet Services and Information Retrieval	Describe major Internet services; create/send email with attachments; research topics using retrieval services.
3.5.5 U6 · L40	Internet Communication and Teleworking	Explain teleworking, telecommuting, videoconferencing, URLs, and cookies; set up and use a teleconferencing platform.
3.5.6 U6 · L41	Integration Activities (Internet & Teleworking)	Internet services and online collaboration practice.
3.5.7 U6 · L42	Electronic Services: E-Commerce, E-Health, E-Government	Define and describe major electronic services with advantages and disadvantages.
3.5.8 U6 · L43	Applications, Benefits & Challenges of E-Services	Compare e-commerce, e-health, and e-government; identify e-government platforms and community problem-solving.
3.5.9 U6 · L44	Computer-Assisted Learning (CAL) and E-Learning	Describe CAL and its role; identify course delivery methods (servers, CD-ROMs, e-books); explain pros and limits.
3.5.10 U6 · L45	Integration Activities (E-Services & E-Learning)	Electronic services and e-learning case studies.

MODULE 04 · 3 CHAPTERS · 18 LESSONS

Cybersecurity, Digital Citizenship, Ethics & Green Computing

Consolidates computer ethics, Cameroon cyber legislation, digital divide, access control, VPNs, intranets/extranets, data validation, malware protection, system recovery, and green computing/e-waste.

CHAPTER 4.1 · 3 LESSONS (CHRONOLOGICAL L6 → U6)

Digital Citizenship, Ethics, Cameroon Law & Digital Divide

Competency: Apply computer ethics, Cameroon legislation, copyright/data protection acts, and reduce the digital divide.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
4.1.1 L6 · L35	**Positive and Negative Uses of Computer Systems**	State positive/negative uses; explain social/economic effects; explain responsible ICT use for communication and culture.
4.1.2 L6 · L36	**Computer Ethics, Legislation and Cameroon Law**	Explain code of ethics and moral obligations; explain unethical acts under Cameroon law; discuss AI deepfakes and algorithmic bias.
4.1.3 L6 · L37	**Data Protection, Copyright and the Digital Divide**	Demonstrate knowledge of data protection and copyright acts; analyse global communication effects and ways to reduce the digital divide.

CHAPTER 4.2 · 12 LESSONS (CHRONOLOGICAL L6 → U6)

Computer & Network Security, Access Control, VPNs & Malware

Competency: Protect systems using CIA principles, firewalls, VPNs, encryption, validation/verification, and malware defense.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
4.2.1 L6 · L38	**Protecting Computer Systems from Illegal Access**	Explain security, reliability, and resilience; describe access controls (passwords, encryption, biometrics, physical security) and backup.
4.2.2 L6 · L39	**System Recovery and Safe Working Practices**	State importance of recovery after failure; explain privacy and safe working practices with examples.
4.2.3 L6 · L40	**Computer Crimes and Combat Measures**	Describe and classify computer-related vs. computer-assisted crimes (copyright infringement, plagiarism); match crimes to combat measures.
4.2.4 L6 · L41	**Malware — Types and Characteristics**	Describe and compare virus, worm, Trojan, rootkit, backdoor, spyware; explain AI in behavioural malware detection and AI-generated attacks.
4.2.5 L6 · L42	**Protecting a Computer System from Malware**	Explain safe practices; scan a system with antivirus; set up protection against unauthorised access using a firewall.
4.2.6 L6 · IA2	**Integration Activity No 2 (Office Task, Law & Security)**	Produce a formatted document and spreadsheet calculation while identifying legal/data-protection obligations and applying security measures.
4.2.7 U6 · L21	**Intranet and Extranet**	Explain intranet and extranet, technologies/requirements to implement them, and contrast their organisational uses.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
4.2.8 U6 · L22	**Virtual Private Networks (VPNs) and Privacy**	Describe VPN principles; explain how protocols ensure privacy; set up a secured VPN using firewalls and digital signatures.
4.2.9 U6 · L23	**Computer and Network Security (CIA Triad)**	Explain confidentiality, integrity, availability; identify security measures (accounts, passwords, firewalls, digital signatures, encryption).
4.2.10 U6 · L24	**Data Protection and Access Control**	Explain encryption and access rights; describe access control methods and importance of protecting CIA.
4.2.11 U6 · L25	**Data Validation and Verification Techniques**	Contrast validation vs. verification; describe validation checks (range, format, length) and verification (visual, double entry).
4.2.12 U6 · L26	**Integration Activities (Network Security)**	Network security and data protection scenarios.

CHAPTER 4.3 · 3 LESSONS (CHRONOLOGICAL L6 → U6)

Green Computing & Responsible E-Waste Disposal

Competency: Apply energy-efficient computing practices and eco-friendly e-waste reuse, recycling, and disposal.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
4.3.1 U6 · L77	**Introduction to Green Computing**	Define green computing and importance; describe energy-efficient practices and identify green vs. non-green habits.
4.3.2 U6 · L78	**Responsible Disposal (E-Waste)**	Explain environmental impact of e-waste; describe eco-friendly disposal, reuse, and recycling; execute community green projects.
4.3.3 U6 · L79	**Integration Activity (Green Computing)**	School/community green computing project.

MODULE 05 · 4 CHAPTERS · 29 LESSONS

Information Systems, SDLC, Databases & IT Project Management

End-to-end systems analysis and project delivery: systems theory, DFDs, IS types, SDLC models, prototyping, changeover, ER modelling, 1NF–3NF normalisation, RDBMS implementation, IT entrepreneurship, WBS, Gantt, PERT, and critical path analysis.

CHAPTER 5.1 · 6 LESSONS (CHRONOLOGICAL L6 → U6)

Systems Theory, Data Flow Diagrams & Information Systems

Competency: Model systems with DFDs, compare TPS/MIS/DSS/EIS, and explain commercial batch vs.

real-time processing.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
5.1.1 L6 · L43	**Types of Systems**	Distinguish natural vs. artificial and manual vs. automatic systems; categorise local systems and state manual system limitations.
5.1.2 L6 · L44	**Modelling a System with a Data Flow Diagram**	Produce a Data Flow Diagram (DFD) for a system; identify an AI-driven system as an automatic system modelled by a DFD.
5.1.3 L6 · L45	**Introduction to Information Systems**	State IS components (people, procedure, technology, data); describe organisational hierarchy; identify IS success/failure factors.
5.1.4 L6 · L46	**Types of Information Systems**	Describe and compare MIS, DSS, EIS, TPS, GIS, HIS, LIS; explain how AI enhances DSS and EIS.
5.1.5 L6 · L47	**General and Commercial Data Processing**	State commercial DP examples; differentiate real-time vs. batch processing; explain how ML extends data processing with pattern prediction.
5.1.6 L6 · IA3	**Integration Activity No 3 (Systems Modelling & AI)**	Model an organisation with a DFD, identify the best IS, and propose automation/robotics/AI improvements.

CHAPTER 5.2 · 4 LESSONS (CHRONOLOGICAL L6 → U6)

System Development Life Cycle (SDLC), Prototyping & Changeover

Competency: Compare SDLC models (Waterfall, Prototyping, Agile, Spiral), AI-assisted prototyping, and changeover strategies.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
5.2.1 L6 · L59	**Introduction to the System Development Life Cycle (SDLC)**	Define the SDLC; describe each stage and identify activities involved at each stage.
5.2.2 L6 · L60	**SDLC Models**	Describe and compare Waterfall, Prototyping, Agile, and Boehm's Spiral models.
5.2.3 L6 · L61	**The Prototyping Model**	Explain prototyping steps, reasons, and types (cost, time, reusability); explain how AI-assisted prototyping accelerates creation.
5.2.4 L6 · L62	**Changeover Strategies**	Explain direct, parallel, pilot, and phased changeover strategies with pros/cons; propose suitable strategies.

CHAPTER 5.3 · 12 LESSONS (CHRONOLOGICAL L6 → U6)

Data Modelling, ER Diagrams, Normalisation & RDBMS

Competency: Design ER diagrams, normalise to 3NF, and build tables, relationships, forms, queries, and

reports in an RDBMS.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
5.3.1 L6 · L63	**Data Modelling Concepts**	Distinguish flat-file vs. relational databases; explain ER, relational, and OO models; describe conceptual, logical, and physical levels.
5.3.2 L6 · L64	**Entity Relationship Model Concepts & Notations**	Explain entity, attribute, PK, FK, relationship, cardinality (1:1, 1:M, M:N); outline Chen and Crow's foot notations.
5.3.3 L6 · L65	**Designing an ER Diagram and Normalisation**	Identify entities, attributes, PKs, and relationships from scenarios; produce ER diagrams (up to 3 entities).
5.3.4 L6 · L66	**Normalisation & Converting ER to Relational Model**	Describe and transform entities into 1NF, 2NF, 3NF; apply rules to convert an ER model into a relational model.
5.3.5 L6 · L67	**Implementing a Relational Model**	Explain DBMS and RDBMS; state examples (Access, Base, MySQL); implement a relational model in an RDBMS.
5.3.6 L6 · L68	**Using an RDBMS for Sorting, Filtering and Queries**	Explain sorting, filtering, and queries; perform sorting/filtering and write/execute queries in an RDBMS.
5.3.7 L6 · IA4	**Integration Activity No 4 (Prototype & ER Model)**	Produce a simple prototype/simulation (with multimedia/VR-AR where relevant) and model its data using an ER diagram.
5.3.8 U6 · L68	**Creating Tables and Relationships (RDBMS)**	Create tables, define constraints, and create relationships between tables using an RDBMS.
5.3.9 U6 · L69	**Queries 1 (RDBMS)**	Perform filtering and sorting of data and write simple queries using an RDBMS.
5.3.10 U6 · L70	**Queries 2 (RDBMS)**	Write advanced/multi-criteria queries using an RDBMS.
5.3.11 U6 · L71	**Reports and Forms (RDBMS)**	Create forms for data input and create reports using an RDBMS.
5.3.12 U6 · L72	**Integration Activity (RDBMS Practical)**	Complete RDBMS tables, relationships, forms, queries, and reports practical.

CHAPTER 5.4 · 7 LESSONS (CHRONOLOGICAL L6 → U6)

IT Entrepreneurship & Project Management (WBS, Gantt, PERT, CPM)

Competency: Identify IT business opportunities and manage projects using WBS, Gantt, PERT, ES/EF/LS/LF, and critical path.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
5.4.1 U6 · L51	**Identifying Business Needs Solvable with IT**	Identify ICT-based businesses and customer needs; discuss IT success stories and detect community needs solvable with IT.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
5.4.2 U6 · L52	**Fundamentals of Project Management 1**	Define project management and importance; describe stage activities, constraints (scope, time, cost), and team roles.
5.4.3 U6 · L53	**Fundamentals of Project Management 2**	Outline project success/failure factors; explain tasks, critical task, ES, EF, slack, lag, lead, and dependencies.
5.4.4 U6 · L54	**Project Planning & Work Breakdown Structure (WBS)**	Explain project planning and WBS; identify/organise project activities and develop a WBS.
5.4.5 U6 · L55	**Project Scheduling Using Gantt and PERT Charts**	Explain Gantt and PERT charts; develop a Gantt schedule, construct a PERT network, and determine project duration.
5.4.6 U6 · L56	**Project Control, Critical Path and Slack Analysis**	Calculate ES, EF, LS, LF; determine the critical path and critical/non-critical tasks; analyse project progress.
5.4.7 U6 · L57	**Integration Activities (Project Management)**	Applied WBS, Gantt, PERT, and critical path project.

MODULE 06 · 2 CHAPTERS · 22 LESSONS

Productivity Tools, Multimedia & Web Development

Practical digital authoring across word processors, advanced spreadsheets (lookup, conditional functions, linking, what-if analysis), multimedia presentations, authoring tools, and full-stack web development (HTML, CSS, JS, CMS, hosting).

CHAPTER 6.1 · 17 LESSONS (CHRONOLOGICAL L6 → U6)

Word Processing, Spreadsheets & Slide Presentations

Competency: Produce professional documents, model data with conditional/lookup/what-if spreadsheets, and deliver multimedia slides.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
6.1.1 L6 · L29	**Editing and Formatting Text**	Perform text editing and formatting in a word processor; identify how AI writing assistants support grammar and style.
6.1.2 L6 · L30	**Editing and Formatting Images and Tables**	Perform editing and formatting of images, graphics, tables, and text boxes.
6.1.3 L6 · L31	**Using Text Boxes and Adjusting Page Layout**	Modify page features (margins, orientation, size, columns); add comments to document sections.
6.1.4 L6 · L32	**Introduction to Spreadsheets**	Describe row, column, cell, address, range, formula, sheet; select ranges/non-contiguous cells; perform basic formatting.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
6.1.5 L6 · L33	**Performing Calculations Using Spreadsheets**	Perform basic arithmetic and functions (SUM, AVERAGE, COUNT, PRODUCT, IF, COUNTIF); explain AI-powered trend detection/forecasting.
6.1.6 L6 · L34	**Types of Cell Referencing and Calculations**	Differentiate relative, absolute, and mixed referencing; solve problems using SUM, AVERAGE, COUNT, PRODUCT, IF, COUNTIF, RANK.
6.1.7 L6 · L54	**Multimedia Systems and Authoring**	Explain multimedia and authoring tools; differentiate multimedia vs. non-multimedia; use an authoring tool to produce short content.
6.1.8 U6 · L58	**Producing Documents Using Word Processing Features**	Apply formatting, reviewing, layout, styles, automatic TOC, and AI-assisted rewriting/summarizing.
6.1.9 U6 · L59	**Formatting and Managing Spreadsheet Data**	Resize rows/columns; format text, currency, %, dates; apply conditional formatting with AI suggestions.
6.1.10 U6 · L60	**Cell Referencing and Basic Formulae**	Perform SUM, AVERAGE, PRODUCT, MAX, MIN; distinguish relative vs. absolute references and apply to ranges.
6.1.11 U6 · L61	**Advanced Functions and Charts**	Apply IF, SUMIF, COUNTIF for financial/statistical analysis; use lookup functions; produce appropriate charts.
6.1.12 U6 · L62	**Linking Worksheets**	Explain how to link worksheets with formulae and determine appropriate formulas for multi-sheet problems.
6.1.13 U6 · L63	**What-If Analysis**	Explain What-If analysis; produce spreadsheet reports and use formula results for projections and decisions.
6.1.14 U6 · L64	**Integration Activities (Advanced Spreadsheets)**	Spreadsheet financial/statistical modeling lab.
6.1.15 U6 · L65	**Producing Slide Presentations**	Explain good presentation practices; select software, apply layouts, and format clear, illustrative slides.
6.1.16 U6 · L66	**Multimedia Presentations, Transitions & Animations**	Add/edit multimedia; apply transitions and animations; apply project management principles to deliver a presentation.
6.1.17 U6 · L67	**Integration Activities (Presentations)**	Multimedia presentation project.

CHAPTER 6.2 · 5 LESSONS (CHRONOLOGICAL L6 → U6)

Web Development & Hosting (HTML, CSS, JavaScript & CMS)

Competency: Build structured, styled, interactive web pages with HTML/CSS/JS and publish via CMS and

hosting.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
6.2.1 U6 · L46	**Developing Simple Web Pages Using HTML**	Identify role of tags and attributes; implement basic HTML tags to structure and create simple web pages.
6.2.2 U6 · L47	**Styling Web Pages with CSS**	Explain CSS purpose; write CSS code to modify appearance and improve layout of a web page.
6.2.3 U6 · L48	**Creating Interactive Web Pages with JavaScript**	Explain JavaScript role; write simple client-side/form-processing JS; integrate HTML, CSS, and JS.
6.2.4 U6 · L49	**Website Hosting and Content Management Systems**	Explain domain names, public IP, DNS, and hosting; use a CMS to create, manage, and publish a website.
6.2.5 U6 · L50	**Integration Activities (Web Development)**	Full interactive website and hosting project.

MODULE 07 · 4 CHAPTERS · 40 LESSONS

Data Structures, Algorithms, Software Design & Programming

Complete software engineering track: data types/structures, software design & UML, algorithms, flowcharts, pseudocode, searching/sorting, dry running, complexity, paradigms, reuse, translators/IDEs, C/Python coding, subroutines, and testing.

CHAPTER 7.1 · 6 LESSONS (CHRONOLOGICAL L6 → U6)

Data Types, Data Structures & Software Design

Competency: Perform operations on arrays, records, stacks, and queues; apply software design methods and UML diagrams.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
7.1.1 L6 · L78	**Data Types and Data Structures**	Identify integer, real, Boolean, character, string; describe arrays, records, stack, and queue; choose appropriate structures.
7.1.2 L6 · L79	**Operations on Data Structures**	Explain element access and perform operations on arrays, records, stack, and queue; combine arrays and records.
7.1.3 L6 · L80	**Approaches to Software Design**	Describe top-down, stepwise refinement, incremental, divide-and-conquer, bottom-up, and modular design; select suitable methods.
7.1.4 L6 · L81	**Applying Software Design Techniques**	Break down problems into smaller units using design techniques; adapt existing solutions to new problems.
7.1.5 L6 · L82	**Representing Software Design**	Explain unit and structure diagrams (class, sequence, composite); describe UML; sketch a structure diagram.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
7.1.6 L6 · IA5	**Integration Activity No 5 (Normalised DB & Logic)**	Convert an ER model to a normalised relational DB and test it; solve binary arithmetic and logic circuit problems.

CHAPTER 7.2 · 13 LESSONS (CHRONOLOGICAL L6 → U6)

Algorithms, Flowcharts, Pseudocode, Searching/Sorting & Dry Running

Competency: Write algorithms with control structures, searching/sorting, dry-run trace tables, and evaluate complexity.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
7.2.1 L6 · L83	**Introduction to Algorithms**	Explain algorithms and characteristics; outline representation methods; compare pros/cons of pseudocode and flowcharts.
7.2.2 L6 · L84	**Algorithmic Instructions**	Describe declarative, input, output, and assignment instructions; describe types of control structures.
7.2.3 L6 · L85	**Representing Algorithms — Flowcharts**	Represent algorithms using standard flowchart symbols for simple real-life problems.
7.2.4 L6 · L86	**Representing Algorithms — Pseudocode**	Write pseudocode for simple problems; distinguish rule-based algorithms from machine-learning algorithms.
7.2.5 L6 · L87	**Control Structures — Sequence and Selection**	Write algorithms using sequence and selection constructs.
7.2.6 L6 · L88	**Control Structures — Loops**	Describe loop types; write algorithms using loops; differentiate between loops and recursion.
7.2.7 L6 · L89	**Sorting and Searching**	Explain sequential and binary search; explain bubble, merge, and insertion sort.
7.2.8 L6 · L90	**Establishing Correctness of Algorithms**	Explain dry running and trace tables; establish algorithm correctness using trace tables.
7.2.9 L6 · L91	**Evaluating Algorithm Performance**	Explain algorithm complexity; compare time efficiency; explain accuracy, precision, and recall for AI/ML algorithms.
7.2.10 U6 · L80	**Selection Control Structure**	Write algorithms that make use of selection control structures.
7.2.11 U6 · L81	**Loop Control Structure**	Write algorithms that make use of loop control structures.
7.2.12 U6 · L82	**Drawing Flowcharts**	Represent algorithms into flowcharts.
7.2.13 U6 · L83	**Dry Running**	Establish the correctness of an algorithm using dry running.

CHAPTER 7.3 · 5 LESSONS (CHRONOLOGICAL L6 → U6)

Programming Paradigms, Software Reuse & Translators/IDEs

Competency: Compare paradigms (procedural, OOP, functional, logic), evaluate software reuse, and explain translators/IDEs.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
7.3.1 L6 · L92	**Programming Paradigms**	Distinguish imperative/procedural, declarative, functional, logic, and OOP paradigms; explain encapsulation and polymorphism.
7.3.2 L6 · L93	**Software Reuse**	Explain internal/external software reuse, libraries, packages, and selection criteria; identify popular AI/ML libraries.
7.3.3 L6 · L94	**Internally vs. Externally Developed Software**	Compare internally vs. externally developed software, outsourcing reuse, and choose between them.
7.3.4 L6 · L95	**Language Translators and IDE**	Differentiate compiler, interpreter, assembler; explain lexical analysis, parsing, code generation, preprocessing; outline IDE features.
7.3.5 L6 · IA6	**Integration Activity No 6 (Algorithm Design & Trace)**	Design a solution with appropriate data structures, write flowchart/pseudocode, trace/test correctness, and refine.

CHAPTER 7.4 · 16 LESSONS (CHRONOLOGICAL L6 → U6)

Coding in C/Python, Subroutines & Software Testing

Competency: Write and debug C/Python programs with control structures, arrays, records, and functions/call stacks; conduct testing.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
7.4.1 L6 · L69	**Types of Software Testing**	Differentiate black-box vs. white-box testing; describe module, integration, system, volume, and acceptance testing.
7.4.2 L6 · L70	**Module Testing**	Explain test case, test data, bug, actual/expected results, test report; carry out module testing; identify AI-powered testing tools.
7.4.3 L6 · L96	**Introduction to Coding**	Explain program, coding, syntax, semantics; install an IDE and explain strategies to learn a programming language.
7.4.4 L6 · L97	**Writing Code 1 (C or Python)**	Identify C/Python program structure; write and debug input, output, and assignment instructions.
7.4.5 L6 · L98	**Writing Code 2 (Selection)**	Write and debug programs in C or Python using selection control structures.
7.4.6 L6 · L99	**Writing Code 3 (Loops)**	Write and debug programs in C or Python using loop control structures.

LESSON	LESSON TITLE	LEARNING OBJECTIVES
7.4.7 L6 · L100	**Writing Code 4 (Arrays)**	Write and debug programs in C or Python using arrays.
7.4.8 L6 · L101	**Writing Code 5 (Arrays & Records)**	Write and debug programs in C or Python using arrays and records.
7.4.9 L6 · L102	**Functions and Procedures 1**	Explain benefits of subroutines; differentiate functions vs. procedures; write and call functions in C or Python.
7.4.10 L6 · L103	**Functions and Procedures 2 (Call Stack)**	Explain stack mechanism for function calls and parameter passing; write/debug functions and dry-run stack frames.
7.4.11 L6 · L104	**Testing a Program 1**	Explain testing steps (code review, static analysis, unit, system); compare debugging techniques (brute force, induction, backtracking).
7.4.12 L6 · L105	**Testing a Program 2**	Test and debug a program using boundary, correct, and erroneous data; produce a test report.
7.4.13 U6 · L84 (1)	**Writing Code 1 (Selection)**	Write programs in C or Python that make use of selection control structures.
7.4.14 U6 · L84 (2)	**Writing Code 2 (Loops)**	Write programs in C or Python that make use of loop control structures.
7.4.15 U6 · L85	**Writing Code 3 (Data Structures)**	Write programs in C or Python that make use of data structures.
7.4.16 U6 · L86	**Integration Activity (Algorithms & Coding)**	Final programming synthesis and problem-solving lab.

SOURCE DOCUMENTS

Official Progression Sheets

1. Republic of Cameroon (MINESEC) — *National Harmonised Progression Sheet for ICT, Lower Sixth* (105 lessons + 6 integration activities).
2. Republic of Cameroon (MINESEC) — *National Harmonised Progression Sheet for ICT, Upper Sixth* (87 lessons).