Lecturers' Handbook

Computer Engineering

Faculty of Engineering, Akpakip Oro University (AOU)

For teaching staff and laboratory supervisors

Draft for departmental approval

1. Purpose and Responsibilities

This handbook supports teaching staff delivering the Computer Engineering programme. It sets out the outcome mapping, teaching schemes, lesson-planning expectations, laboratory procedures, assessment design and marking standards. It is a working guide and does not replace the university's regulations.

- Deliver courses against the learning outcomes in the Students' Handbook.
- Publish the week-by-week scheme and assessment plan to students in the first week.
- Design assessments that test the stated outcomes at the appropriate level.
- Mark consistently, moderate with colleagues and give timely, specific feedback.
- Supervise projects, SIWES placements and laboratory work to the standards set here.

2. Programme Outcome Mapping

Each graduate competency is developed across several courses. The mapping below is generated from the course list and must be checked against the CCMAS mapping before accreditation.

Competency	Main contributing courses
Programming and software	CPE 104, CPE 111, CPE 205, CPE 213, CPE 304, CPE 313, CPE 504
Digital and computer architecture	CPE 202, CPE 204, CPE 307, CPE 210, CPE 414, CPE 112
Electronics	CPE 203, CPE 208, CPE 115, CPE 110
Algorithms and data	CPE 201, CPE 306, CPE 212, CPE 312, CPE 405, CPE 413
Operating systems and real-time systems	CPE 209, CPE 303, CPE 310, CPE 404, CPE 407
Embedded systems	CPE 211, CPE 305, CPE 301, CPE 406, CPE 314, CPE 501, CPE 411
Networks and communications	CPE 302, CPE 309, CPE 311, CPE 402, CPE 401, CPE 511, CPE 502, CPE 410
Security and reliability	CPE 403, CPE 409, CPE 412, CPE 513, CPE 514
Design and integration	CPE 509, CPE 515
Professional practice	CPE 106, CPE 105, CPE 416, CPE 512, CPE 508

3. Teaching Schemes

For each semester, the table shows the delivery mode and assessment method. Contact hours and rooms are set by the timetable.

Year 1: Engineering Foundations

Semester 1

Code	Course	Units	Delivery	Assessment
CPE 101	Mathematics I	3	Lecture	CA 30 / Exam 70
CPE 102	Discrete Mathematics	3	Lecture	CA 30 / Exam 70
CPE 103	Introduction to Computing	2	Lecture	CA 60 / Exam 40
CPE 104	Programming I	3	Lecture	CA 40 / Exam 60
CPE 105	Communication in English	2	Lecture	CA 60 / Exam 40
CPE 106	Introductory Engineering	2	Lecture	CA 50 / Exam 50
CPE 107	Engineering Laboratory Practicals	1	Lecture with lab / practical	Lab reports 100
CPE 108	Introduction to Computer Engineering	2	Lecture	CA 40 / Exam 60

Semester 2

Code	Course	Units	Delivery	Assessment
CPE 109	Mathematics II	3	Lecture	CA 30 / Exam 70
CPE 110	Physics for Electronics	3	Lecture	CA 30 / Exam 70
CPE 111	Programming II	3	Lecture	CA 40 / Exam 60
CPE 112	Computer Hardware Fundamentals	2	Lecture	CA 40 / Exam 60
CPE 113	Nigerian Peoples and Cultures	2	Lecture	CA 40 / Exam 60
CPE 114	Engineering Workshop and Safety	2	Lecture with lab / practical	Practical 70 / Exam 30
CPE 115	Electrical Principles	3	Lecture	CA 30 / Exam 70
CPE 116	Computing Laboratory I	1	Lecture with lab / practical	Lab reports 100

Year 2: Core Engineering

Semester 3

Code	Course	Units	Delivery	Assessment
CPE 201	Data Structures and Algorithms	3	Lecture	CA 30 / Exam 70
CPE 202	Digital Logic Design	3	Lecture	CA 40 / Exam 60
CPE 203	Electronic Devices I	3	Lecture	CA 30 / Exam 70
CPE 204	Computer Organisation and Architecture	3	Lecture	CA 30 / Exam 70
CPE 205	Object-Oriented Programming	3	Lecture	CA 40 / Exam 60
CPE 206	Engineering Mathematics	3	Lecture	CA 30 / Exam 70
CPE 207	Probability and Statistics	2	Lecture	CA 40 / Exam 60
CPE 208	Circuit Theory I	3	Lecture	CA 30 / Exam 70

Semester 4

Code	Course	Units	Delivery	Assessment
CPE 209	Operating Systems	3	Lecture	CA 30 / Exam 70
CPE 210	Digital Electronics	3	Lecture	CA 40 / Exam 60
CPE 211	Microprocessors	3	Lecture	CA 40 / Exam 60
CPE 212	Database Systems	3	Lecture	CA 40 / Exam 60
CPE 213	Software Engineering I	3	Lecture	CA 40 / Exam 60
CPE 214	Signals and Systems	3	Lecture	CA 30 / Exam 70
CPE 215	Computing Laboratory II	2	Lecture with lab / practical	Lab reports 100
CPE 216	Numerical Methods	2	Lecture	CA 40 / Exam 60

Year 3: Specialist Foundations

Semester 5

Code	Course	Units	Delivery	Assessment
CPE 301	Embedded Systems Design	3	Lecture	CA 40 / Exam 60
CPE 302	Computer Networks I	3	Lecture	CA 30 / Exam 70
CPE 303	Real-Time Systems	3	Lecture	CA 30 / Exam 70
CPE 304	Software Engineering II	3	Lecture	CA 40 / Exam 60
CPE 305	Microcontroller Interfacing	3	Lecture	CA 40 / Exam 60

Code	Course	Units	Delivery	Assessment
CPE 306	Algorithms and Complexity	3	Lecture	CA 30 / Exam 70
CPE 307	Computer Architecture II	3	Lecture	CA 30 / Exam 70
CPE 308	Engineering Statistics and Research Methods	2	Lecture	CA 40 / Exam 60

Semester 6

Code	Course	Units	Delivery	Assessment
CPE 309	Computer Networks II	3	Lecture	CA 30 / Exam 70
CPE 310	Embedded Linux and Device Drivers	3	Lecture	CA 40 / Exam 60
CPE 311	Network Programming	3	Lecture	CA 40 / Exam 60
CPE 312	Compiler Design	3	Lecture	CA 30 / Exam 70
CPE 313	Mobile Application Development	3	Lecture	CA 40 / Exam 60
CPE 314	Sensors and Actuators Interfacing	2	Lecture	CA 40 / Exam 60
CPE 315	Computing Laboratory III	2	Lecture with lab / practical	Lab reports 100
CPE 316	Engineering Economics	2	Lecture	CA 40 / Exam 60

Year 4: Advanced Design and Industrial Practice

Semester 7

Code	Course	Units	Delivery	Assessment
CPE 401	Internet of Things	3	Lecture	CA 30 / Exam 70
CPE 402	Wireless Sensor Networks	3	Lecture	CA 30 / Exam 70
CPE 403	Network Security	3	Lecture	CA 30 / Exam 70
CPE 404	Distributed Systems	3	Lecture	CA 30 / Exam 70
CPE 405	Machine Learning Fundamentals	3	Lecture	CA 30 / Exam 70
CPE 406	Hardware-Software Co-design	3	Lecture	CA 40 / Exam 60
CPE 407	Cloud Computing	2	Lecture	CA 40 / Exam 60
CPE 408	Research Project I	2	Studio / project supervision	Proposal 100

Semester 8

Code	Course	Units	Delivery	Assessment
CPE 409	Embedded Systems Security	2	Lecture	CA 40 / Exam 60
CPE 410	Network Design and Administration	3	Lecture	CA 30 / Exam 70
CPE 411	Industrial Embedded Systems	3	Lecture	CA 30 / Exam 70
CPE 412	Cybersecurity Operations	3	Lecture	CA 30 / Exam 70
CPE 413	Artificial Intelligence	3	Lecture	CA 30 / Exam 70
CPE 414	Digital Systems Verification	2	Lecture	CA 40 / Exam 60
CPE 415	Systems Design Standards	2	Lecture	CA 40 / Exam 60
CPE 416	Industrial Training (SIWES)	3	Industrial placement	Logbook 40 / Report 40 / Assessor 20

Year 5: Specialisation, Integration and Professional Readiness

Semester 9

Code	Course	Units	Delivery	Assessment
CPE 501	Advanced Embedded Systems	3	Lecture	CA 30 / Exam 70
CPE 502	Advanced Network Systems	3	Lecture	CA 30 / Exam 70
CPE 503	Edge Computing and Embedded AI	2	Lecture	CA 40 / Exam 60
CPE 504	DevOps and Software Delivery	2	Lecture	CA 40 / Exam 60
CPE 505	Research Project II	3	Studio / project supervision	Project 100
CPE 506	Engineering Project Management	2	Studio / project supervision	CA 40 / Exam 60
CPE 507	Technical Elective I	2	Lecture	CA 40 / Exam 60
CPE 508	Intellectual Property and IT Law	2	Lecture	CA 50 / Exam 50

Semester 10

Code	Course	Units	Delivery	Assessment
CPE 509	Integrated Computing Systems Design	3	Lecture	Design project 100
CPE 510	Embedded Product Development	2	Lecture	CA 40 / Exam 60
CPE 511	Enterprise Network Deployment	2	Lecture	CA 40 / Exam 60
CPE 512	Engineering Professional Practice and Ethics	2	Lecture	CA 50 / Exam 50
CPE 513	Quality Assurance and Testing	2	Lecture	CA 40 / Exam 60
CPE 514	Risk and Safety in Computing Systems	2	Lecture	CA 40 / Exam 60
CPE 515	Final-Year Design Project and Defence	4	Studio / project supervision	Project 60 / Defence 40
CPE 516	Technical Elective II	2	Lecture	CA 40 / Exam 60

4. Lesson Planning

Every lecture and practical session should have a written lesson plan, kept with the course file and sampled during moderation.

Lesson plan template

- **Course and week:** code, title, week number.
- **Learning outcome(s):** the outcome(s) this session addresses.
- **Prerequisites, resources and activities:** what students must know, what is needed, and the timed outline.
- **Check for understanding:** at least one formative question or task.
- **Homework or preparation:** the task and its due date.

Worked example: Combinational circuit design (2-hour session)

Time	Activity	Check for understanding
0 to 10 min	Starter: recall of Boolean identities and truth tables	Quick oral questions
10 to 40 min	Explanation: sum-of-products and product-of-sums design	Students write the SOP form for a given truth table
40 to 65 min	Worked example: design a 2-bit adder from logic gates	Pairs design a 1-bit full adder
65 to 95 min	Simulation exercise: build and test the adder in a simulator	Each pair reports its test results
95 to 110 min	Short quiz on the session content	Marks recorded for formative tracking
110 to 120 min	Summary; homework on a 4-bit comparator	Homework due next session

5. Laboratory and Practical Instructions

Standard references are given for orientation. Confirm the current edition and the department's approved procedure before issuing instructions to students.

5.1 Combinational and sequential logic (Digital Logic Design)

Aim. Design, build and test a combinational circuit and a synchronous counter.

Equipment.

- Logic trainer board
- Logic gate ICs
- Flip-flop ICs
- Logic probe and multimeter

Procedure.

1. Derive the minimised Boolean expression for the specified function.
2. Build the circuit and verify the truth table.
3. Build a 3-bit synchronous counter from flip-flops and verify its sequence.
4. Record the results and any faults found.

Safety.

- Check supply polarity and voltage before connecting ICs.
- Power down before changing connections.

Report requirements. Include the minimised expressions, the truth tables, the timing observations and the fault log.

5.2 GPIO, timers and interrupts (Microcontroller Interfacing)

Aim. Program a microcontroller to read a button, drive an LED and handle a timer interrupt.

Equipment.

- Microcontroller development board
- LEDs, push button and resistors
- Programmer or debugger
- IDE on a PC

Procedure.

1. Configure one pin as input with pull-up and one pin as output.
2. Write the code to toggle the LED on each button press, with debouncing.
3. Configure a timer to generate an interrupt at a fixed interval.
4. Test the program and measure the interrupt period with an oscilloscope or logic analyser.

Safety.

- Use the correct current-limiting resistors for LEDs.
- Keep the board on an anti-static mat.

Report requirements. Include the code, the configuration settings, the measured interrupt period and the debouncing method used.

5.3 Packet capture and protocol analysis (Computer Networks I)

Aim. Capture traffic on an authorised network and identify protocol headers.

Equipment.

- Lab PC with packet capture software
- Isolated lab network
- Managed switch
- Authorisation sheet

Procedure.

1. Confirm written authorisation and the scope of the capture.
2. Start a capture while generating traffic with a simple ping and a web request.
3. Identify the Ethernet, IP, and TCP or UDP headers in the captured frames.
4. Filter the capture and record the fields for one TCP handshake.

Safety.

- Capture only on the isolated lab network or with written authorisation.
- Do not record or share personal data from captures.

Report requirements. Include the capture summary, annotated header fields, the TCP handshake sequence and an ethics statement.

5.4 Sensor data acquisition over UART (Sensors and Actuators Interfacing)

Aim. Read an analogue sensor with an ADC and transmit the readings to a PC over UART.

Equipment.

- Microcontroller board with ADC
- Analogue sensor (for example a thermistor)
- USB-serial adaptor and PC
- Terminal software

Procedure.

1. Wire the sensor and its signal conditioning to the ADC input.
2. Program the ADC read and the UART transmission.
3. Calibrate the sensor against a reference reading.
4. Log the readings on the PC and check their accuracy and noise.

Safety.

- Confirm the ADC input voltage range before connection.
- Never connect the board to mains-powered equipment.

Report requirements. Include the circuit diagram, calibration data, the logged results and an analysis of noise and accuracy.

6. Assessment Design

6.1 Assessment blueprint for written examinations

Cognitive level	Typical weight	Typical question form
Remember	10%	Definitions, standard values, short recall
Understand	20%	Explain principles, interpret results
Apply	40%	Calculations, applying standards and methods
Analyse	20%	Multi-step problems and system analysis
Evaluate	10%	Design choices, critique, justification

6.2 Continuous assessment

- Each course should have at least two CA components, spread across the semester.
- At least one CA item should test an outcome the exam does not.
- Publish all CA marking schemes before the task is issued.

7. Rubrics

Each criterion is scored from 1 to 4. The total is converted to the mark scale in the regulations.

7.1 Design project rubric

Criterion	4 Excellent	3 Good	2 Satisfactory	1 Unsatisfactory
Design concept	Clear, justified and suited to the brief	Sound with minor gaps	Partly justified	Unclear or unsuitable
Calculations or analysis	Complete, correct and checked	Minor errors, method correct	Method partly correct	Major errors or missing
Use of standards	Correct standards applied and cited	Mostly applied correctly	Some misapplied	Not used
Drawings, code or specifications	Clear, complete, to convention	Minor omissions	Incomplete	Not usable
Safety and sustainability	Addressed in depth	Addressed adequately	Mentioned only	Not addressed

7.2 Laboratory report rubric

Criterion	4 Excellent	3 Good	2 Satisfactory	1 Unsatisfactory
Method	Complete and reproducible	Mostly complete	Partly complete	Not usable
Data and analysis	Accurate and correctly processed	Minor errors	Several errors	Major errors
Discussion	Insightful, error sources analysed	Adequate	Superficial	Absent
Presentation	Clear and professional	Mostly clear	Hard to follow	Poor

7.3 Oral defence rubric

Criterion	4 Excellent	3 Good	2 Satisfactory	1 Unsatisfactory
Explanation	Clear and complete	Clear with minor gaps	Partial	Unclear
Response to questions	Confident, accurate, reasoned	Mostly accurate	Some accurate	Mostly inaccurate
Professional judgement	Justifies trade-offs well	Justifies most decisions	Limited	None

8. Marking, Moderation and Feedback

- Mark against the published scheme, and record the reason for any deduction.
- Moderate a sample of at least 10% of scripts or reports across the mark range, with a second marker.
- Return feedback within the period set by the regulations, identifying strengths and next steps.
- Record any change to a published mark, with the reason and the approver.

9. Supervision of Projects and SIWES

- Supervisors meet each student at least every two weeks and keep a meeting log.
- Projects require an approved written proposal before execution.

- Check each SIWES placement before the student starts; review the logbook and report, and complete the assessor's form.

10. Quality Assurance and Accreditation Readiness

- Keep a course file for each course: specification, scheme, assessment items, marking schemes, moderation records and feedback.
- Maintain records of equipment, calibration and maintenance for each laboratory.
- Invite an external examiner to review standards and assessment, and act on their report.
- Update the CCMAS mapping whenever a course changes.

11. Academic Integrity and Professional Ethics

- Explain the university's academic integrity policy at the start of every course.
- Report suspected misconduct through the department to the appropriate committee.
- Model professional ethics, including honest reporting of data and clear acknowledgement of sources.