UNIVERSITY of CAMBRIDGE
International Examinations

GCE Advanced Level Computing
Scheme of Work

Paper 3
Systems Software Mechanisms, Machine Architecture, Database Theory, Programming Paradigms and Integrated Information Systems

Introduction

This section provides candidates with knowledge and understanding of the following aspects of computer systems:

- the functions of operating systems
- the functions and purposes of translators
- computer architectures and the fetch-execute cycle
- data representation, data structures and data manipulation
- programming paradigms
- databases
- use of systems and data
- systems development, implementation, management and applications
- simulation and real-time processing
- common network environments, connectivity and security issues

Recommended Prior Knowledge

Candidates should have studied Paper One.

Important note

Centres should deliver Paper Four (Computing Project) alongside this Paper.

Scheme of Work

Session Plan 101 – Functions of Operating Systems

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.1**	3.1.1 Features of operating systems	• describe the main features of operating systems

Classroom Exercises	Notes
Review Operating System work for Paper One concentrating on the characteristics of: • Single-User • Multi-User • Network Systems Introduce the features of Operating Systems that support multi-users and networking • Memory Management • Scheduling • Distributed Systems	Include: • Memory Management • Scheduling • Distributed systems

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.1**	3.1.2 Scheduling 3.1.4 Job Queues & Priorities	• define and explain the purpose of scheduling, job queues, priorities and how they are used to manage job throughput

Classroom Exercises

Introduce the concepts of jobs, processes and scheduling.

Define the terms

- job
- job queue
- priorities (including the concepts of processor bound and peripheral bound)
- process (including running, runnable and suspended states)
- scheduling

Introduce scheduling and discuss the following benefits:

- maximise use of hardware resources
- maximise throughput
- allocate resources fairly to all users
- provide acceptable response time for interactive users
- provide acceptable turnaround time for batch users
- manage system performance (e.g. temporarily increase time taken to respond if the system is overloaded)
- prevent deadlock

Use simple diagrams to show the benefits of scheduling e.g.

Notes

Include the following scheduling algorithms

- shortest job first, shortest remaining time, round robin

<table>
<tr><th>Assessment Objectives</th><th>Performance Criteria</th><th>Classroom Ideas</th></tr>
<tr><td>• 3.1</td><td>3.1.3
Interrupt Handling</td><td>• explain how interrupts are used to obtain processor time and how processing of interrupted jobs may later be resumed</td></tr>
<tr><th colspan="2">Classroom Exercises</th><th>Notes</th></tr>
<tr><td colspan="2">Define the term interrupt (a signal from some device/source seeking the attention of the processor), the different classes of interrupt and the need to assign different priorities to interrupts (so that when two interrupts occur at the same time or an interrupt occurs whilst another is being serviced, the interrupt with the highest priority is dealt with first). Classes of interrupt should include:
• Hardware failure Highest Priority
• Program
• Timer
• I/O Lowest Priority

Realise that the current program is also assigned a priority</td><td rowspan="2">Typical sources of interrupts should be identified including the following classes:

• program generated
• processor time generated
• hardware failure

Vectored interrupt handling

May be described with some students</td></tr>
<tr><td colspan="2">Introduce concept of interrupt service routines and outline the sequence of actions

1. save status (registers etc.)
2. determine cause (poll status flags)
3. take relevant action
4. restore status

Possibly explain, using diagrams on the board, the use of vectors to determine the location in memory of the appropriate routine.</td></tr>
</table>

Session Plan 102 – Memory Management, PC and Network Operating Systems

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.1**	3.1.7 Modern personal computer operating systems	• describe the main components of a typical desktop PC operating system • describe the main components of a network operating system

Classroom Exercises	Notes
Define the terms boot file, a file containing commands to automatically configure a personal computer on start up, and file allocation table (FAT), a list held on disk by an operating system to maintain and manage disk space used for file storage. Use classroom discussion to identify the components of a personal computer operating system, as students should have experience of using at least one operating system.	Include: • use of the file allocation table • purpose of the boot file Include: • transparency • directory services • security • network printing (including an understanding of spooling)

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.1**	3.1.5 Memory management	• explain how memory is managed in a typical modern computer system

Classroom Exercises	Notes
Define the following terms: • virtual memory (include the reasons for use e.g. allows more processes to be run than could be held in main memory) • paging • segmentation Using diagrams on the board (or pre-prepared as a handout), explain the operation of segmentation and paging in virtual memory systems, highlighting the differences between the two systems.	Include: • virtual memory • paging • segmentation

Session Plan 103a – Functions and Purposes of Translators

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.2**	3.2.1 Types of Translator 3.2.2 Lexical Analysis 3.2.3 Syntax Analysis 3.2.4 Code Generation 3.2.4 Linkers and Loaders	• describe the difference between interpretation and compilation • describe what happens during lexical analysis • describe what happens during syntax analysis, explaining how errors are handled • explain the code generation phase • explain the purpose of linkers and loaders

Classroom Exercises

Review types of translator (Paper One Session Six) for High-level languages and the conversion of source code to object code. Extend this to highlight the differences between compilation and interpretation including at a minimum:

- compiler translates the whole program (source code) into object code that can be stored and re-used
- interpreter translates and executes a program line by line. No object code is stored for further use, a program has to be translated each time it is used

Discuss the advantages and disadvantages of compilation and interpretation highlighting when it would be appropriate to use a compiler or an interpreter (e.g. use an interpreter during program development as errors can be easily checked and modified). As students have used translators they should be able to contribute to a discussion.

Introduce the stages of compilation:

- lexical analysis
- syntax analysis
- code generation
- linking and loading

Describe in general terms what happens during each phase including tokenisation, the use of the symbol table and handling errors. Use sample code from a programming language that your students are familiar with to demonstrate the general principles.

Resources

- review translators from Paper One
- worksheet to re-enforce knowledge – perhaps using examples from a familiar High Level Language

Notes

Include:
- source code
- object code

Session Plan 103b – Computer Architectures and the Fetch-Execute Cycle

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.3**	3.3.1 Von Neumann architecture 3.3.2 Registers: purpose and use	• describe basic Von Neumann architecture, identifying the need for, and the uses of, special registers in the functioning of a processor

Classroom Exercises

Introduce the concept of Von Neumann architecture – any computer that takes a single instruction then obeys it before processing the next instruction.

Describe the contents and the use of the following registers:

- Sequence Control Register
- Current Instruction Register
- Memory Address Register
- Memory Buffer Register (Memory Data Register)
- Accumulator

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.3**	3.3.3 Fetch-execute cycle	• describe, in simple terms, the fetch/decode/ execute/reset cycle, and the effects of the stages of the cycle on specific registers.

Classroom Exercises

Prepare a diagram showing the flow of data/instructions through the registers. Include the use of Data/Address/Control buses. If possible provide a demonstration of the fetch-execute cycle using one of the computer programs commercially available and/or search for and use one of the demonstrations available on the world-wide-web. Using a set of simple Assembly Language/Machine Code instructions trace the contents of each of the registers, this can be done as a whole class exercise giving the opportunity to work through the cycle several times using different types of instruction.

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.3**	3.3.4 Parallel processors	• discuss parallel processor systems, their uses, advantages and disadvantages

Classroom Exercises

Define parallel processing (the simultaneous use of several processors to perform a single job). Compare this to the Von Neumann computer.

Provide mini pre-determined scenarios of the use of parallel processing e.g. weather forecasting, processing live images from a satellite, artificial intelligence.

Session Plan 104 – Data Representation – Number Systems

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.4**	3.4.1 Number systems	• express numbers in binary coded decimal (BCD), octal and hexadecimal • describe and use two's complement and sign and magnitude to represent positive and negative integers • perform integer binary arithmetic: addition and subtraction

Classroom Exercises

Revise the use of binary numbers. Extend this work to include octal (base 8), hexadecimal (base 16) and Binary Coded Decimal (BCD).

Provide students with a worksheet containing codes in binary, octal, hexadecimal and BCD to be converted into denary. Also provide conversions from denary values in all three number bases and BCD (include how many bytes would be required).

Demonstrate, with board work, the use of two's complement and sign and magnitude to represent positive and negative numbers. Stress how to represent both positive and negative numbers because many students often only consider the use of negative numbers.

Introduce addition and subtraction using two's complement for integers.

Provide a worksheet with practice questions converting positive and negative denary integers to two's complement and sign and magnitude and addition and subtraction of the binary integers. Provide questions that given the number of bits available (e.g. 1 byte, 2 bytes etc). This will allow for discussion of overflow.

Resources

- prepared questions for students to attempt with model answers

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.4**	3.4.2 Floating point binary	• demonstrate an understanding of floating point representation of a real binary number

Notes

Include definitions of :

- Mantissa
- Exponent
- Overflow
- Underflow

} Both to be stored in two's complement form

Resources

- prepared examples of decimal numbers and their floating point representation

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.4**	3.4.3 Normalisation of floating binary numbers	• normalise a real binary number • discuss the trade-off between accuracy and range when representing numbers

Classroom Exercises

Explain the structure of a floating-point number, including definitions of the mantissa (non-zero fractional part) and exponent (integer power). Provide examples showing the range of values that can be stored and how a normalised number allows for the greatest precision for a given size of mantissa. Explain how the increase in range leads to a decrease in precision and introduce the ideas of underflow (exponent too small) or overflow (exponent too large) as the result of a calculation.

Use method of

- change to a binary number
- normalise the binary value
- adjust the exponent to accept the normalisation

to create floating point representations.

Set worksheet exercises to practise the conversion of a decimal number to binary floating point and binary floating-point numbers to decimal. Include positive and negative numbers, large numbers and fractional values.

Resources

- prepared questions for students to attempt with model answers

Session Plan 105 – Data Representation – Data Structures

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.4**	3.4.4 Implementation of data structures	• explain the difference between static and dynamic implementation of data structures, highlighting the advantages and disadvantages of each • describe algorithms for the insertion, deletion and amendment of data items stored in linked-list, stack and queue structures • describe insertion, deletion and amendment of data items in a tree structure

Classroom Exercises	Notes
Revise the purpose of and structures of arrays, stacks, queues (including the concepts of LIFO, FIFO, stack pointers, pushing and popping to and from stacks and queues) and linked lists. Discuss the advantages and disadvantages of static and dynamic data structures. Introduce the tree data structure using simple diagrams It is appropriate to provide an algorithm say for insertion of data in a queue and let the students write their own algorithm for deletion before introducing the model answer.	Include the following data structures: • linked lists • stacks • queues • trees Note that students should not attempt to learn algorithms. The deletion of values from a tree is not expected. Diagrams should be used to explain/describe the amendments of the named data structures. Ensure that students include checks for error conditions in their responses.

Session Plan 106 – Data Representation – Data Manipulation

<table>
<tr><th>Assessment Objectives</th><th>Performance Criteria</th><th colspan="2">Classroom Ideas</th></tr>
<tr><td>• 3.4</td><td>3.4.4
Searching and sorting</td><td colspan="2">• explain the difference between binary searching and serial searching, highlighting the advantages and disadvantages of each
• explain the difference between insertion sort and merge sort
• describe algorithms for implementing insertion sort and merge sort methods</td></tr>
<tr><th colspan="3">Classroom Exercises</th><th>Notes</th></tr>
<tr><td colspan="3">Demonstrate the use serial and binary searches with several sets of data. Choose the data sets very carefully to show the advantages and disadvantages of each type of search by using both algorithms on the same set of data.

Demonstrate the following sort routines:

• insertion sort
• merge sort

Start with the insertion sort, as it is the easiest one to understand.</td><td>Note that here, as in most cases within the course; the emphasis is on describing an algorithm rather than on any particular method used for describing algorithms.</td></tr>
</table>

Resources

- algorithms for
  - binary searching
  - serial searching

Session Plan 107 – Programming Paradigms

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.5**	3.5.1 Types of languages and typical applications	• describe with the aid of examples, the characteristics of a variety of programming paradigms • explain the terms object-oriented, declarative and procedural as applied to high-level languages

Classroom Exercises	Notes
Introduce the different types of High Level programming languages. Provide definitions of the following types of programming languages and the characteristics of each: • declarative • imperative • procedural • object oriented • low level	Include the following paradigms: • low level • procedural • declarative • object-oriented Knowledge of the syntax of programming languages is not required.

Assessment Objectives	Performance Criteria	Classroom Ideas
• 3.5	3.5.2 General features of procedural languages	• explain how functions, procedures and their related variables may be used to develop a program in a structured way, using stepwise refinement • describe the use of parameters, local and global variables as standard programming techniques • explain how a stack is used to handle procedure calling and parameter passing

Classroom Exercises

Review top down approach, procedures and functions and introduce stepwise refinement. Describe the use of global variables, local variables and parameter passing (by value and by reference).

Review functions of a stack, explain the use of a stack to handle procedure calling and return including pushing of return address, parameter values/addresses on entry to a procedure and popping of same on exit from a procedure.

Session Plan 108 – Procedural, Declarative, Functional and Object Oriented Languages

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.5**	3.5.2 Features of object-oriented languages	• discuss the concepts and, using examples, show an understanding of data encapsulation, classes and derived classes, and inheritance

Classroom Exercises	Notes
Explain the concepts of object-oriented languages including at a minimum: • encapsulation (keeping together data structures and methods) • classes • derived classes • inheritance (derived classes carry the data structures and methods of the superclass) Use everyday examples to introduce these ideas e.g. class definition of clock, derived classes – analogue clock and digital clock.	

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.5**	3.5.2 Features of declarative languages	• discuss the concepts and, using examples, show an understanding of backtracking, instantiation and satisfying goals

Classroom Exercises
Explain the concepts of declarative languages including at a minimum: • rules • facts • backtracking • instantiation (binding of a variable to a value during resolution, lasting only long enough to satisfy one complete goal) • satisfying goals Use everyday examples to introduce these ideas.

Session Plan 109 – Features of Low level Languages, Generations of Programming Language and Syntax Definition

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.5**	3.5.2 Features of low level languages	• explain the concepts of addressing of memory

Classroom Exercises	Notes
Using simplified Assembly Language describe the following ways of addressing memory: • direct (using the contents of the address) • indirect (using the contents of the address as a pointer to another address) • indexed (using the contents of the address in combination with the contents of an index register to determine the address) This could be demonstrated by the use of a simple set of examples on the board or the use of a commercially available simulation program depending upon the resources available.	Must include: • direct • indirect • indexed Only concepts required here, as students are not expected to be able to write low-level language code.

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.5**	3.5.3 Methods for defining syntax	• explain the need for, and be able to apply, BNF (Backus-Naur form) and syntax diagrams

Classroom Exercises

Demonstrate on the board the use of Backus-Naur form (BNF) as a formal method to describe simple syntax of a programming language.

Use the following meta symbols:
::= is defined by
| OR
<> meta variable

E.g.
<hexdigit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F

Student centred exercise using worksheets to reinforce / test knowledge – perhaps providing simple examples to extend. Revise the answers to the worksheet as a class discussion to reinforce the concepts studied.

Demonstrate the use of syntax diagrams as a formal method to describe simple syntax of a set of rules.

Resources

- worksheets with exercises in defining syntax rules using both methods

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.5**	3.5.2 Generations of Programming Language	• using examples, describe the nature and purpose of 3rd and 4th generation languages

Classroom Exercises	Notes
Third generation languages are those high level languages that use a structure syntax. Fourth generation languages offer a more powerful language which can decide on a course of action to produce a desired result.	

Session Plan 110 – Database Structures, Normalisation and E-R Modelling

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.6**	3.6.1 Database design	• describe flat files and databases • explain the advantages that using a relational database gives over flat files

Classroom Exercises

Review work done on files, indexing and key fields from Paper 1. Contrast this with the use of databases:

Teachers should concentrate on relations databases, it being enough to simply mention network and hierarchical and how they differ.

Use diagrams to show each type of database.

Introduce the advantages of using a relational database rather than a flat file including:

- data independence
- data consistency
- lack of duplication of data
- less redundant data

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.6**	3.6.2 Normalisation and data modelling	• design simple relational databases to the third normal form (3NF)

Classroom Exercises	Notes
Using a practical example of a previously set up relational database introduce the concepts of: • tables • primary keys • foreign keys • secondary keys • views of data Demonstrate and explain the purpose of each of these concepts using the pre-prepared database then introduce the students to the formally set out underlying data structures. e.g. Table_loan (loan_no, bookno, libmemno, borrowdate, expreturndate, actreturndate) Where loan_no is the primary key of the loan table Bookno and libmemno are foreign keys from other tables in a library database. --- Demonstrate the principles of normalisation starting with a flat file data structure and working through the stages of normalisation: • 1st normal form – remove repeating data • 2nd normal form – remove partial key dependencies • 3rd normal form – remove non key dependencies Choose your examples very carefully to ensure the one used for demonstration and the first few that the students attempt need work to be done at all stages (many examples may not yield composite keys so there can be no partial key dependencies). Provide pre-determined scenarios e.g. customer orders, student records etc. that allow the students to identify specify and normalise the data structures required.	The practical use of a relational database management system such as ACCESS to allow the students to develop their own database would reinforce these concepts and the use of forms, DDL, DML and access rights.

Resources

- worksheets providing different data structures to be normalised

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.6**	3.6.2 Normalisation and data modelling	• draw entity-relationship (E-R) diagrams to represent diagrammatically the data model

Classroom Exercises

Introduce the concepts of entities and relationships (one to one, one to many, many to many). Use everyday occurrences to demonstrate these concepts e.g. the student teacher model can be discussed showing the idea of a many to many relationship between student and teacher and how the introduction of other entities such as class meeting can help organise the model.

Explain how the relationships need to be carefully labelled in order to show understanding. Similar data structures can be used to the ones prepared for the normalisation exercise, this will help enforce how these two techniques complement each other.

Ensure familiarity with the concept of a link entity.

Session Plan 111 – Database Structures and Management

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.6**	3.6.3 Methods and tools for analysing and implementing database design	• design forms for input, deletion, modification and querying of a database • describe the structure of a database management system (DBMS)

Classroom Exercises	Notes
Use a practical exercise to review the concepts of good HCI design from Paper 1 and design forms for: • data entry • data deletion • data modification • querying data	
Referring to the practical work completed introduce the main functions of a DBMS: • Data Dictionary (an internal file containing the name, description, characteristics, relationships for each data item and information about programs and users. • Data Description/Definition Language (DDL) • Data Manipulation Language (DML) Explain that this information is stored with the data in a database system. Students may have used a GUI to define and manipulate data but a demonstration of the underlying commands actually used (e.g. showing the SQL commands produced by a QBE query) could be used to show the functions of a DDL and a DML as SQL has both properties.	Include the function and purpose of: • Data Dictionary • Data Description Language (DDL) • Data Manipulation Language (DML)

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.6**	3.6.4 Control of access to relational database elements	• explain the importance of varying the access allowed to database elements at different times and for different categories of user

Classroom Exercises	Notes
Extend the discussion on DBMS to include the use of different views of the system for different categories of users and at different times. Discuss the type of access e.g. read, read/write etc; the view allowed e.g. different types of user could be allowed access to certain elements of data; the effect of time on availability of data e.g. some elements not yet released or some elements may have been archived. A pre-prepared database and/or pre determined scenario would help students understand these concepts e.g. a database of students, courses, and examination results or chose another example that the students will be able to relate to.	Include the following types of access: • read data • read/write data • design chapter

Session Plan 112 – Use of Systems and Data

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.7**	3.7.1 The commercial value of data	• identify data that has commercial value; explaining why such data has this value and discuss contemporary trends in the compilation and use of valuable databases

Classroom Exercises

Use pre determined scenarios to illustrate the commercial value of data. Current articles from the computer press could be used to provide examples that highlight the following:

- commercial value of data (and the need for it to be relevant, from a reliable source, kept up to date etc.)

Compilation and use of databases

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.7**	3.7.2 The importance of standards	• explain the advantages of standardisation and describe some areas of standardisation such as file formats, ISDN, OSI model and its use together with communications protocols.

Classroom Exercises	Notes
Describe how the use of standards has aided computerisation by using examples of current de facto (wide use has led to market domination e.g. Windows) and de jure (pre-defined industry standards e.g. OSI model). Discuss in class the advantages of the following standards and also use this to review the work on networks from Paper 1: • file formats • ISDN • OSI model and communications protocol	Include: • file formats • ISDN • OSI model • communications protocols

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.7**	3.7.3 Communications and electronic commerce	• describe ways in which computers aid communication • identify situations in which the transmission of data has created/could create new opportunities for businesses and individuals to conduct

Classroom Exercises	Notes
Describe the use of computers in communications, including as a minimum: • voicemail • email • digital telephone facilities • Internet use by e-commerce • tele/video conferencing Ensure that a discussion about how e-commerce works is included e.g. use a flow diagram to describe the process of buying goods on-line.	Include: • voicemail • email • digital telephone system facilities • e-commerce over the Internet • tele/videoconferencing

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.7**	3.7.5 Training	• identify and describe training and re-training requirements for a given situation.

Classroom Exercises

Use pre determined scenarios (try and include both critical and non-critical systems) to illustrate both long-term and short-term changes that occur when a computerised system is introduced, consider patterns of work and quality of output. Use these scenarios to identify training and retraining requirements. Discussion of the requirements should develop a sound understanding of both these concepts.

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.7**	3.7.6 Effects of introducing systems	• describe the substantial changes which occur as a result of introducing computing systems

Notes

Include short-term and long-term changes:

- in patterns of work
- in quality of output

Session Plan 113 – Systems Development and Implementation

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.8**	3.8.1 Methodologies and software tools for system development	• identify and describe how the use of methodologies/ techniques and software tools for developing computer systems aid the systems analyst/designer and programmer in terms of the documentation, step-by-step logical progression through tasks and cross-checking mechanisms

Classroom Exercises	Notes
Discuss the ways that a system can be decomposed using a variety of methods including: • Data Flow Diagrams (flow of data through system) • E-R Modelling (Identification of data objects, their structure and the relationships between them) • Task Modelling Show how formal methods can be used to document and crosscheck the above. Also discuss the benefits that abstraction (using different kinds of model to concentrate on one aspect of the system at a time) bring to the system developer.	Include: • E-R Modelling • Data Flow Diagrams • Gantt charts • Critical Path Analysis

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.8**	3.8.3 3.8.4 Application types and technical requirements	• discuss the technical requirements of a system necessary to implement a range of different computer applications • explain the need to provide appropriate response times for different applications and its implications for hardware, software and data structures

Classroom Exercises	Notes
Use pre prepared scenarios (try and include applications requiring different response times) to discuss the technical requirements of a computer system, including: • hardware • data structures • operating systems • communications • interface software • other utility software	Include: • hardware • operating systems • communications • interface software • other utility software

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.8**	3.8.5 Choice of implementation approaches	• select, plan and justify appropriate implementation approaches for a range of different applications

Classroom Exercises	Notes
Review work from Paper 1 on implementation approaches; provide extra pre determined scenarios for the students to take decisions on. Discuss the justification for each decision taken. Ensure that the students understand the differences between different approaches to implementation and that they can explain which is most appropriate in given circumstances.	Study these approaches: • direct • parallel • phased • pilot

Session Plan 114 – Project and Systems Management

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.8**	3.8.6 Systems management and monitoring	• discuss the implications of managing, monitoring and maintenance of systems

Classroom Exercises	Notes
Discuss the management, monitoring and maintenance of working systems including at a minimum: • the need for quality control and management and the use of appropriate tools • the need for up-to-date documentation of the system • the benefits and implications of the use of a Software Audit • the implications of any hardware updates Again the use of pre determined scenarios and classroom discussion will benefit students' understanding of these concepts.	The following should be included: • up-to-date documentation • software audit • quality control and management • hardware updates

Session Plan 115 – Real Time Applications

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.9**	3.9.1 3.9.2 Applications of real-time computing	• describe real-time applications

Classroom Exercises

Describe a variety of real time systems stressing the need for speed of response to external events but also including the need for reliability and recovery.

Introduce the idea of a feedback loop by describing a simple system e.g. a temperature control system attached to a heater and a fan. Also discuss the need for sensors and actuators to implement this system. Extend this work to look at a variety of other real time systems that use the following types of signals:

- visible
- tactile
- audible
- other physical signals

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.9**	3.9.3 The use of robots	• discuss the use of robots in a variety of situations

Classroom Exercises	Notes
Extend the discussion on sensors to include robots. Make use of pre prepared scenarios to stimulate discussion of the use of robots in manufacturing (e.g. high precision jobs such as painting, welding and riveting) and hazardous environments e.g. (cleaning toxic waste or bomb disposal).	Cover these situations: • manufacturing • hazardous environments

Session Plan 116 – Simulation and Parallel Processing

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.9**	3.9.4 Uses of simulation	• explain the reasons for simulation, such as to change time-scales and/or save costs and/or avoid danger • describe the uses of simulation to assist in design, to make predictions, to test hypotheses
Classroom Exercises		
Look at the different reasons for modelling different types of situation: • predictions e.g. weather forecasting • design e.g. testing stresses in bridge design • hypotheses e.g. a country's economics over varying time scales and conditions Discuss the importance of observing the effect of the variable elements in any simulation and also simulation limitations where there are unpredictable, random events e.g. the effect of a coup d'etat on a country's economy, very bad weather in flight simulators.		

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.9**	3.9.6 Processing requirements	• explain the large processing requirements of some systems and hence recognise the need for parallel architectures

Session Plan 117 – Data Transmission, Network Components and Environments

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.10**	3.10.1 Data transmission	• describe methods used to organise LANs and WANs • describe typical rates of data transmission associated with different topologies and methods • describe different media for transmitting data and their carrying capabilities

Classroom Exercises

Review work done on networking in Paper One Session 12. Provide two pre determined scenarios one of a large LAN that contains several smaller LANs and one of a WAN showing the following elements. (Large diagrams showing elements, topology, transmission rates etc. would be useful):

- topology
- connections (include the media used and possible/actual transmission rates)
- routing (include the use of switches, bridges, routers, modems/terminal adaptors and their effects on the transmission rates)
- protocols (also discuss effects on transmission rates)
- network operating systems

Student centred exercise using worksheets to reinforce/test knowledge – perhaps providing simple scenarios for the students to design a suitable network and provide justification for the components chosen. Revise the answers to the worksheet as a class discussion to reinforce the concepts studied.

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.10**	3.10.2 3.10.3 Communications and electronic commerce	• select and justify an appropriate network system for a particular application

Classroom Exercises	Notes
Review work on networks from Paper 1. Elements required to build a network (e.g. hardware, cabling, bandwidth etc); the students can select and justify the appropriate elements required for each network.	Include bandwidth required to transmit different forms of data: • text • sound • real-time sampled data and Video • time-sensitive data Hardware required: • file servers • hubs • repeaters • switches

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.10**	3.10.2 Network components	• explain the different purposes of network components

Notes

Include the following:
- switches
- routers
- bridges
- modems

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.10**	3.10.4 Common network environments	• discuss common network environments, their facilities, structure and ability to exchange information using appropriate software and techniques • describe how a network environment affects the user interface provided • describe the facilities provided by electronic mail systems (including voicemail) • explain that distribution of a network can have implications both for data and responsibility

Classroom Exercises	Notes
Define open networks, referring to previous work on standards and networks. As students will probably have used open networking systems e.g. the Internet identify the facilities available and the advantages offered. Discuss the exchange of information over networks, perhaps compare and contrast the use of different methods e.g. FTP (File Transfer Protocol) with the use of HTTP for multimedia.	Use the internet as an example of an open network and intranets as examples of closed networks.

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.10**	3.10.7	

Describe and demonstrate the use of email including the following facilities: • composing • responding • filing • copying • attaching • sending on • multiple recipients	Include: • composing • responding • filing • copying • attaching • sending on • multiple recipients The practical use of electronic mail would be useful.

Session Plan 118 – Hypertext Linking Systems, Confidentiality, Encryption and Authentication

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.10**	3.10.5 3.10.6 Use of networks to support hyperlinking systems such as the world wide web (WWW)	• describe the purpose of hypertext linking, identifying the means by which it can be achieved such as hotwords/links, buttons and hypertext mark-up language (HTML) • describe the basic features of mark-up languages

Classroom Exercises	Notes
Demonstrate the features of a basic web page including the following: • use of tags (including on/off pairing, e.g. <HTML>…..</HTML>, <HEAD>…..</HEAD>, <BODY>….</BODY> etc) • links (within a page, to another page, to another web site) • methods of linking (hotwords, buttons, hotspots etc.) • basic formatting (centring, text size, colours etc.)	Practical preparation of a simple web page using HTML using notepad or similar may reinforce the concepts.

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.10**	3.10.8 Issues of confidentiality	• discuss the problem of maintaining confidentiality of data on an open network and how to address this problem

Classroom Exercises

Discuss the problems of ensuring the confidentiality of data as it is being transferred across and stored at nodes on an open network, where coding and transmission methods are freely available. Include the following ideas in your discussion:

- prevention of access to data when stored e.g. physical security, use of access levels and passwords
- protection of data, from malicious interference, during transmission e.g. use of encryption (including public and private keys), screening of cables, problems with radio transmission, benefits of packet switching etc.
- ensuring that information is from a trusted source e.g. use of digital certificates to verify the authenticity of the message sender and provide the receiver with the means to encode a reply
- use of authorisation techniques to ensure that confidential information only reaches the intended recipient e.g. use of passwords, responses to special questions, provision of memorable data etc.

Assessment Objectives	Performance Criteria	Classroom Ideas
• **3.10**	3.10.6 Encryption and authentication techniques	• explain the need for encryption, authorisation and authentication techniques

Notes

- students will not be expected to know any specific method in detail but will be expected to understand the needs for them.