UNIVERSITY OF CAMBRIDGE INTERNATIONAL EXAMINATIONS

Cambridge International Diploma in Computing
Advanced Level

Scheme of Work

5216
Computer Systems, Communications and Software
Core Module

UNIVERSITY of CAMBRIDGE
International Examinations

Introduction

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

- components of a Computer System and Modes of Use
- system software
- programming tools and techniques
- data: their representation, structure and management
- hardware
- data transmission and networking

The systems development life cycle is studied with reference to particular applications. Therefore, candidates are expected to look at a range of different types of application areas. Although candidates are not expected to have specific knowledge of every one, candidates should be able to make use of relevant examples for the purpose of illustration. This section also provides candidates with understanding of the following aspects of computer systems:

- systems development life cycle
- choosing applications software for application areas
- handling of data in information systems
- characteristics of information systems
- implications of computer use

Tutor Preparation Required to Start This Module

- decide on the programming language to be used ensuring that candidates can fulfill all the syllabus requirements with the selected package
- install this software so that it is accessible to all students
- provide instructions showing what the students have to do
- prepare a bank of appropriate supplementary resources such as work done by students in previous years, brochures, catalogues, worksheets to test students knowledge at each stage

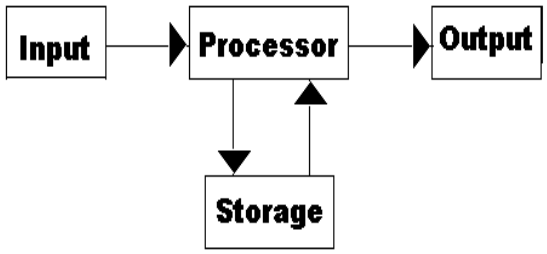
Important note:

Some centres may wish to deliver elements of Module two (Structured Practical Tasks) alongside this module. To help centres who wish to use this approach, the relevant Session Plans and Performance Criteria for the theory work on Module one are mapped for each element listed in Module two.

Scheme of Work

Session Plan One – Hardware, Software and Modes of Use

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.1**	1.1.1 Types of Hardware	Define the terms associated with computer hardware, describe how the elements of the system relate to one another and practise classification of devices.

Classroom Exercises	Notes
Show students structure diagram:  Define all four elements. Show students devices, pictures of devices or other resources and identify by name and type. Define storage devices and storage media. Students given a worksheet containing pictures of a range of input, output and storage devices. For each they have to name, state the type of device, its purpose and at least one common application for the device. Mark the worksheets as an oral exercise to stimulate discussion particularly on common applications. This exercise makes a good ice breaker for new students because many can relate some devices to their own experiences. See additional notes in session plan one above. Briefly discuss software and its relationship with hardware.	• sometimes useful to have devices like old floppy discs (which have been opened), CD ROMs, printers etc so that students have visual images of a range of devices • 'describe the purpose of each device' can include attributes like: - for input devices types of data captured, which can lead into suitable applications - for output devices speed of device, purchase costs, running costs, which can lead into suitable applications - for storage devices, access speed, storage capacity, costs

Resources

- prepared structure diagram
- samples of / pictures of different devices and storage media
- worksheet containing pictures of a range of input, output and storage devices; to test the name, type, purpose, and one common application for each picture

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.1**	1.1.2 Types of Software	• define the different types of software • describe the purpose of each of the software types giving common examples of applications for each type

Classroom Exercises	Notes
Demonstrate the use of a computer; using any example, even booting the computer or logging onto a network. Question the students: What is happening? How is it happening? How does the computer know what to do? Lead into discussion on software and its relationship with hardware, emphasising the fact that without software a computer system could not function. Define software – sets of instructions to make a computer do something are grouped together as programs. These programs make up the software of a computer system. Ask the students to think about types of software they have encountered and develop a list of software types and functions. For each give examples of common software names (and product names) which the students will be using. Include notes on Systems software and which of the above categories form part of this label. Discuss the difference between generic and product names, drawing parallels with common examples from everyday life. (e.g. 'car' rather than 'Fiat'). Prepared worksheet to allow students to research and/or to test knowledge, e.g. filling in the missing words from a given list of software types.	Minimum software lists: • Operating System (OS) • User Interface (including GUI) • Translator • Utilities • Programming languages • Generic / Common Applications Ensure that students learn generic names like database software rather than product names like 'Access' and explain that only generic names will be accepted in the examination. (A fuller discussion about the meaning of 'generic' packages takes place later in the course).

Resources

- worksheet to test knowledge – perhaps filling in the missing words from a given list of software types

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.1**	1.1.1 Types of Hardware 1.1.2 Types of Software	Discuss modes of computer use and the interrelationship between applications and modes of use.

Classroom Exercises	Notes
Define modes of computer use and suggest a range of applications. Ask the students to work out which applications would be best suited to a given mode of computer use. The type of computer use (e.g. multi-user) relates closely to the modes of use: demonstrate the interdependence of the way the computer is used and the mode of use. For example, batch work is likely do be done off-line. Short student centred exercise using worksheets to research / reinforce / test knowledge – perhaps filling in the missing words in a series of questions about applications from a list of the modes.	Minimum list: • Batch • Real-time • On-line • Off-line Other useful terms: • Single-User • Multi-User • Network Systems

Resources

- a list of descriptions of applications exemplifying the different ways computers are used
- worksheet to test knowledge – perhaps filling in the missing words in a series of questions about applications from a list of the modes

Session Plan Two – System Software

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.2**	1.2.1 Operating Systems	• discuss the purpose of operating systems • describe the characteristics of different types of operating systems • describe the uses of different types of operating systems

Classroom Exercises	Notes
Provide screenshots demonstrating a number of different operating system environments; for example Windows, DOS, Linux and Unix. Ask the students to discuss the similarities and differences. Define operating system – a set of programs designed to run in the background on a computer system, giving an environment in which application software can be executed. Question the students: What are operating systems for (remembering the examples you have seen and worked with)? What can all operating systems do? Describe the characteristics of different types of operating systems. Describe the uses of different types of operating systems and relate the work to the different modes of computer use covered in previous sessions. Reinforce the discussion about the purpose of operating systems with handouts or notes. Finish with a short exercise using worksheets to research / reinforce / test knowledge – mapping the characteristics of different operating systems to their uses and even commonly used OS environment names.	Include: • Batch • Real-time • Single-User • Multi-User • Network Systems

Resources

- screenshots or working examples (on computers) of different operating systems
- handouts about operating systems if required
- worksheet to test knowledge – perhaps mapping the characteristics of different operating systems to their uses and even commonly used OS environment names

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.2**	1.2.2 User Interfaces	• describe the characteristics of different types of user interfaces • discuss the types of user interfaces which make them appropriate for use by different types of users

Classroom Exercises	Notes
Using demonstration materials from the previous session showing different types of operating systems, illustrate the differences between graphical and command line interfaces. Ask students to propose appropriate names for the different types, and steer them towards the correct names. Discuss the types of user interfaces which make them appropriate for use by different types of users. Lead the discussion with questions such as: Why do many people dislike command line interfaces? Who would use command line interfaces – and why? What skills do users need to operate a graphical interface like Windows? Reinforce the class discussion with notes or handouts describing the characteristics of different types of user interfaces	Include: • GUI (WIMP) • menus • icons • forms • natural language • command line

Resources

- screenshots or working examples (on computers) of different operating systems
- handouts about user interfaces if required
- worksheet to test knowledge – perhaps listing characteristics of each type of user interface and mapping these to a range of types of end user

Session Plan Three – Programming Tools and Techniques

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.3**	1.3.1 Problem -solving techniques	• turn a problem into a mathematical formula • turn a problem into a series of stages (algorithm) • turn an algorithm into a flow diagram • discuss turning the flow diagram into a program

Classroom Exercises

Introduce problem solving techniques by turning verbal descriptions into mathematical algorithms (words to formulae)
e.g. area of a rectangle = LxB
perimeter of a rectangle = L+L+B+B = 2L + 2B = 2(L+B).
Use diagrams to demonstrate how the formulae are built up.

Use the example about perimeters to show that the same solution can be written in different ways, and this should stimulate discussion about the efficiency of algorithms to solve problems.

Other suitable questions could be area of a triangle = ½ BH.
Break down the mathematical algorithm into steps:

Find Base
Find Height
Multiply Base by Height
Divide Answer by 2

Ask students to suggest how to turn these steps into a flow diagram. Compare methods and notation to highlight the need for a convention to use in flow diagrams.

Introduce correct flow diagram notation and flow lines down and to the right. All lines against the flow must be annotated as such.

Student exercises could include:
Write algorithm and draw flow diagram for:

- calculation of the area of a rectangle
- calculation of the perimeter of a rectangle
- (an advanced one) – make a cup of tea or coffee. (Discuss the need to think about what is needed as input)

Discuss turning a flow diagram into a program (sequence of instructions), giving examples of the type of instructions they will be using. Ensure that the later exercises involve the translation of algorithm into programs which can be functional and (later) tested.

Resources

- prepared answers for different algorithms and flow diagrams

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.3**	1.3.2 Features of procedural programming language (I)	• introduce a top down approach and modularity • introduce the concept of program sequence

Classroom Exercises

Review last session on algorithms and flow diagrams.

Discuss how to find the area of a 'house' made up from a square and a triangle by working out the area of the triangle, working out the area of the square and then adding the two together.

Use this to explain what a top down approach is – a large complex problem broken into smaller more manageable pieces. When each of the smaller problems has been solved then all the pieces are put together to give an overall solution.
Introduce concept of modularity.

Now discuss the problem of controlling a robotic production line. The problem is complex, but can be divided into smaller manageable pieces:

- how is the data going to be collected from sensors and stored in the system?
- what operations need to be processed and in what order?
- how is the decision going to be made about when to perform each operation?
- what outputs are necessary, and how are they controlled?

More than one person or team of people can be engaged on solving different parts of the same problem at the same time. Therefore the problem can be solved more quickly.

Give a similar problem to four ‘teams’ in the classroom. The problem is to design a new computerised traffic light system for (name a local set of highway traffic lights controlling a road junction). Identify the four areas to be addressed as discussed in the production line example.

Give each group time to brainstorm a solution, put all solutions together and see if that fulfils the original task. In this instance it does not matter if the group's solutions work – if not it is better to provoke discussion about definition of each group's task, what we asked them to do, what input they required and what output they were expected to give.

This should develop the idea of modular notation (on input, process, on output) as used in standard programming techniques.

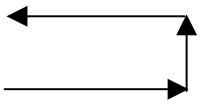
Introduce program sequence. Give the students small exercises of putting these things into the correct order to produce a required output. For this use small programs (four to six lines in length) with each line of code printed on a card and the students have to place the cards in order. Written exercises like this may be used for homework.

Resources

- prepared breakdown of each problem to be set
- potential answers to each module’s problem
- small programs – cut up onto cards (see below)
- worksheets on program sequencing and logical sequencing

Session Plan Four – Programming Tools and Techniques

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.3**	1.3.2 Features of procedural programming language (II)	Use everyday example to introduce the ideas of: • decisions/selection with Y/N solutions • If..Then..Else constructs • Repetition

Classroom Exercises	Notes
Produce the algorithm to make a cup of tea (or coffee) from a previous session. Remind students about how to draw flow diagrams and ask them to attempt to draw a flow diagram to show how to make a cup of tea. This will lead to discussions about selection, sequence and repetition. **Examples** **decisions/selection with Y/N solutions** Use from cup of tea: Do you take sugar? Discuss framing the questions to always give Yes or No answers. Create a flow diagram to illustrate these steps. **IF..Then..Else constructs** Use from cup of tea: Do you take sugar? If Yes then go to section which adds sugar to the cup, else If No go to the section for milk. Create a further flow diagram for this section (perhaps as a module called Sugar). **Repetition** Use from cup of tea in the Sugar module: Add a little sugar Is this enough? If Yes return from the module If not go back to Add a little sugar  Use other everyday examples to demonstrate the need for the following further programming constructs: • Repeat …Until • While … Endwhile • For … Next A number of smaller exercises need to be developed here to reinforce these elements – perhaps set as homework. Ensure that the later exercises involve the translation of algorithm into programs which can be functional and (later) tested.	introduce all elements from flow diagrams repetition include: • Repeat …Until • While … Endwhile • For … Next Examples could include: • checking a number of items through a checkout (using repeat ….. until or while …. Endwhile) • adding five consecutive numbers together (using for … next)

Session Plan Five – Programming Tools and Techniques

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.3**	1.3.2 Features of procedural programming language (III)	• define and introduce procedures and functions • reinforce structure • reinforce good practice relating to annotation of code and variable naming

Classroom Exercises	Notes
Discuss the features of a procedural programming language, ensuring that the definitions of procedures and functions are understood. A procedure which gives a value at the end is known as a function. Refer to Session Three and expand the notion of modularity by introducing procedures and functions. Explain idea of referencing every module with on entry, process, on exit and passing of variables. Stress the importance of noting what has changed (become corrupted) as a result of using the module. At this stage it is worth focusing students' attention on good practice, particularly in relation to the annotation of code. Introduce indented code for loops, annotation to explain each section and the use of meaningful variable names that would allow another programmer to debug or edit the code. Some examples of well annotated code could be compared to one with no annotation. Some ideas for programs involving many of the skills developed so far are:- • write a program to sort shapes, input data and to calculate the area and perimeters of those shapes. Shapes to be sorted would include rectangles, squares, triangles and circles • write a problem to show the times tables (multiplication table), where the user inputs the table that is to be produced as well as the starting number and finishing number for that table	

Resources

- examples of annotated code
- examples of code with no annotation
- prepared questions for students to attempt with model answers

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.3**	1.3.3 Basic translation process	• introduce the two types of translator • discuss the process of source-object code

Classroom Exercises	Notes
Introduce the idea that programs written in high-level languages are translated into machine code, which is executed directly on the computer. This is the basic translation process and all the source code is translated into object code. Include an explanation of the difference between a compiler and an interpreter, demonstrating, if appropriate, with examples. It may be particularly effective to include examples with errors: with compiled languages the errors will be found before the program is run at all, but with an interpreted language the program will begin to run and produce an error message when the mistake is encountered. Emphasise the fact that the object code is normally much bigger than the source code because it takes lots of machine code commands for each of the high level language commands.	C++, COBOL, Visual Basic 6 and Pascal are all compiled languages. Lisp, Perl and Basic are interpreted languages.

Session Plan Six – Programming Tools and Techniques

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.3**	1.3.4 Program testing	• explain the three major error types which will need to be tested • introduce the concept of developing a test plan and in the production of error free programs • introduce black box testing • introduce white box testing • introduce concept of Alpha tests and Beta tests • relate all testing to final programming project

Classroom Exercises	Notes
Introduce the idea of Program Testing. Programs must be tested carefully in order to ensure that they work as planned. Give the students examples of code containing different types of errors and ask the students to determine the three main errors which are commonly made - syntax errors, logical errors and arithmetic errors. (Arithmetic errors are a subset of logical errors). Introduce the concept of a test plan and ensure that test plans cover all the three major error types. Introduce the idea of Black box testing: Black-box test design treats the system as a 'black-box', so it does not explicitly use knowledge of the internal code and structure. Black-box test design is usually described as focusing on testing functional requirements, external specifications or interface specifications of the program or module. The ultimate goal for black-box testing is to test every possible combination and value of input. Introduce white box testing – testing all routes through a program. Give the students a number of small programs, with test plans which they should classify as black box or white box testing. They must annotate the test plan to identify the routes tested and all variables used, ensuring that functionality is correct. (Keep the programs simple but multi-routed). Ask the students to identify which programs contain errors, the nature of the error, fix the error and retest. Ensure that the students get into the habit of annotating all tests within the test plan. Introduce the concepts of Alpha testing and Beta testing. These are user tests. Explain that the programmer tests focus on error-free processing. User tests focus on usability, functionality, and performance. User testing with test data is called Alpha testing. This is then followed by Beta testing during which users use the system with their own data. If appropriate at this stage, relate this to the programming project which must include a full test plan, black box, white box, alpha and beta testing.	Emphasise that testing is not a casual thing and that test plans should enable programmers to create completely error free programs. For black box testing, students should be shown how to select inputs which are normal, error bounds, erroneous and abnormal. As an example for black box testing, use the following: E.g.:- Problem: Read two numbers, 'a' and 'b'. Put the larger of the numbers into the box 'c'. Conditions to be tested: • both numbers positive - 'a' larger - 'b' larger • one number positive - 'a' positive - 'b' positive • both numbers negative - 'a' larger (less negative) - 'b' larger • one number zero - 'a' = 0 - 'b' = 0 • both numbers equal - both positive - both negative - both zero • other conditions...

Resources

- a number of small programs containing errors and test plans with pre determined data. There needs to be two sets with different types of error to allow for both black box and white box testing

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.3**	1.3.5 Program maintenance	• introduce translator diagnostics • introduce and demonstrate cross referencing, program traces and variable dumps • introduce and demonstrate dry runs (desk checks)

Classroom Exercises

Describe the use of translator diagnostics to assist in program debugging. This is possibly best achieved with a simulated 'live' program test/s.

Demonstrate the use of cross-referencing, program traces and variable dumps to identify where possible errors occur.

Demonstrate the use of dry runs (desk checking) on simple arithmetic programs with loops, these can then be demonstrated using traces or variable dumps and the results compared.

Students can load and test pre prepared programs using all of the above techniques and error correct where appropriate.

Resources

- prepared programs containing errors to be used for a simulated 'live' test
- prepared programs which require cross-referencing, program traces and variable dumps to fix bugs
- simple arithmetic programs with loops to enable dry runs (desk checking) to be performed
- test files for students to load and test using all the above techniques
- trace tables for dry run (desk checking)

Session Plan Seven – Data Representation

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.4**	1.4.2 Data Types	• describe binary and hexadecimal • convert simple numbers between denary and binary and hexadecimal • describe character format data types, explain the use of these codes

Classroom Exercises	Notes
Explain that computers store information in Binary format i.e. base 2. Discuss the concept of other number bases, counting in binary (base 2) and hexadecimal (base 16). Teach conversion to and from each of these number bases and denary (base 10). When introducing binary ensure that students cover bits, bytes (nibbles) and words. Introduce character sets and teach conversion using tables and both binary and hex codes. Students should be given a worksheet containing codes in denary, binary and hexadecimal to convert into both ASCII and EBCDIC. Also conversions back to numeric values in all three number bases	Include Least and Most Significant Bits and Status and Boolean Variables **ASCII** American Standard Code for Information Interchange **BCD** Binary Coded Decimal **EBCDIC** Extended Binary Coded Decimal Interchange Code

Resources

- sample exercises for conversion denary-binary-hexadecimal with model answers
- ASCII and EBCDIC tables
- exercise to translate hexadecimal into ASCII and EBCDIC and from character set into hexadecimal

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.4**	1.4.1 Number Systems 1.4.2 Data Types	• explain and use different data structures. Explain which data structures are suitable for which types of data • explain relative storage sizes of different data types giving advantages and disadvantages of each type in specific situations

Classroom Exercises	Notes
Explain the features of and difference between different data structures. Identify suitable data structures for different functions. Explain which data structures are suitable for which types of data. Explain relative storage sizes of different data types giving advantages and disadvantages of each type in specific situations. Give students a worksheet to select the correct data structures for different samples of data. Enhance this to include storage sizes. Marking these worksheets orally in class should provoke and stimulate discussion on different storage types and the relative merits of each for specific functions. Ensure that all data structures listed are covered.	• structures must include: Text / Character Integer (Numeric) Floating point (Numeric) Character / Text Boolean (Y/N True/False) Date/Time Currency

Resources

- worksheet/s to select the correct data structures for different samples of data

Session Plan Eight – Data Structure

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.4**	1.4.3 Data Structures	• explain the purpose and structure of single-dimensional and multi-dimensional arrays • explain memory allocation, initialising arrays and reading data into arrays • practise setting up arrays, in one and two dimensions and reading data into these arrays • design and write routine/s to perform a simple serial search on an array

Classroom Exercises	Notes
Demonstrate the purpose of an array using an example. Explain the purpose and structure of single-dimensional arrays. Explain memory allocation, initialising arrays and reading data into arrays. Set worksheet exercises to practise setting up single dimensional arrays and reading data into these arrays. As a class activity or in small groups – design and write routine/s to perform a simple serial search on an array. Use a further example to demonstrate the need for multi-dimensional arrays and give students similar exercises to work on single-dimensional arrays. Discuss the need for dimensioning arrays and demonstrate how to do this. **Extension activity** This idea can be extended to develop into a small programming project, as it not only develops the use of arrays in programming but is ideally suited to practise for testing – in particular dry runs (desk checks). The pre determined data from the worksheets could be used as the test data.	A suitable example of a program needing an array would be to add two fractions. An array holds the numerator and denominator of each fraction, the lowest common denominator and the numerator and denominator of the sum of the two fractions. A possible activity for work on multi-dimensional arrays: searching on one variable from a multi-dimensional array and returning the value of another variable.

Resources

- examples to be used for demonstration purposes
- worksheets with data to be read into arrays – some single-dimensional and some multi-dimensional. Include data suitable for designing a simple serial search on an array, perhaps searching on one variable from a multi-dimensional array and returning the value of another variable

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.4**	1.4.3 Data Structures	• explain the purpose and structure of single pointer linked lists for solving simple problems • explain initialising and setting up linked lists • perform a simple serial search on a linked list

Classroom Exercises	Notes
Introduce the idea of linked lists using a suitable example and relate the use of lists to arrays, highlighting the difference between a list and an array. Identify the value of linked lists in memory saving and having lists of indeterminate length. Explain initialising and setting up linked lists, this may be best done using diagrams. Demonstrate performing searches on linked lists using manual methods. Perform simple serial searches on linked lists from the worksheet/s which should contain data that can be manually linked to give students exercises in producing linked lists. More than one copy of the same data that can be linked in different ways would be useful.	An array is a single piece of memory, but a linked list contains as many pieces of memory as there are items in the list. Detailed algorithms for the searches are not expected.

Resources

- an example to demonstrate the need for a linked list
- worksheets with data that can be manually linked to give students exercises in producing linked lists. More than one copy of the same data that can be linked in different ways would be useful

Session Plan Nine – Data Structure

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.4**	1.4.3 Data Structures	Explain the purpose of, and structures of stacks and queues and the concepts of: • First In First Out (FIFO) • Last In First Out (LIFO) • stack pointers

Classroom Exercises	Notes
Use a diagram (perhaps on a white board with moveable magnetic flags as start pointers and end pointers – if one is available) to explain what a stack is, discussing the concept of LIFO. Ensure that students have a sound understanding of the concepts before progressing to the concept of a queue. Through questioning develop the idea of the stack pointer. Demonstrate examples of questions involving pushing and popping to and from stacks and queues. Students attempt worksheets with questions on pushing and pulling to and from stacks and queues. The marking of these questions may be better as a class discussion to reinforce the concepts studied.	Mention deques which allow insertions and deletions at either end.

Resources

- diagram to be used for demonstration
- worksheets with questions on pushing and pulling to and from stacks and queues

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.4**	1.4.3 Data Structures	• explain concepts of files, records, fields • explain simple linear file structure – with a variety of fields of fixed length • explain the function of indexing and key fields • calculate from this structure an estimated file size given the number of records

Classroom Exercises	Notes
1. Explain concepts of files, records, fields and items of data. One way of introducing this is to describe a large database of which the students have some knowledge (e.g. a large vehicle database used to register all cars in the country or a database holding personal details of all the clients of a large company). Discuss files which are applications package based and files which are designed to be used with a bespoke program. Explain simple linear file structures using a variety of fields of fixed length. Use a localised database with data with which (preferably) the students are familiar. A good example might be a small database holding data about the students in this particular class (without holding data which may be too personal – like addresses or age). Examine the data structures used and discuss why each was selected for the purpose. For example: using a Y/N field called Male rather than a gender field requiring six bytes for 'Female'. Explain the function of indexing and key fields with reference to speed of access. From a worksheet which contains details of data to be stored in a system, design a data structure for an applications package, and for a programming language. Discuss the answers to the worksheet as a class discussion to reinforce the concepts studied. 2. Revise 1K = 1024 = $2^{8.}$ It may be necessary to do some work on mathematical techniques of estimation and approximation prior to starting this exercise. Demonstrate how to calculate from a pre- defined file structure an estimated file size given the number of records. Students attempt worksheets with questions on calculating estimated file sizes from given data structures and number of records. Marking these questions may be better as a class discussion to reinforce the concepts studied.	• discuss application based files • discuss program based files • calculations should include work on approximation and rounding

Resources

- worksheets with candidates to develop file structures for given sets of data where the students calculate the appropriate field length
- worksheets with given sets of data – must include fixed length fields where the students calculate the appropriate field length. They must also calculate the estimated file sizes given the same data structure and the number of records

Session Plan Ten – Data Structure

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.4**	1.4.3 Data Structures	• discuss serial, sequential, indexed sequential and random access searches • explain the purpose and operation of hashing algorithms

Classroom Exercises	Notes
Introduce the idea of different access methods for stored data and different data structures. Relate the every day examples such as tape recorders and CD players. Cover serial access, sequential access, indexed sequential and random access. Compare and contrast the different types of searches, discussing advantages and disadvantages of each. Define the terms and describe the use of indexes and the calculation of hashing algorithms.	For sequential access, a good method of delivery is to demonstrate the process of adding a record to a file (and the need to move all other data down one record). This will require pre prepared worksheets. For indexed sequential, use the examples of bank accounts, or UK post codes to demonstrate first level, second level…… indexes. There is no need to go into the detailed calculations of the hashing algorithms.

Resources

- prepared worksheets showing data structures which the students can refer to in the demonstration/s

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.4**	1.4.3 Data Structures	• discuss the need for selecting data types for a given problem and list the advantages and disadvantages of the different structures prior to justifying the selection made

Classroom Exercises	Notes
Present a range of examples and ask students to identify appropriate data types. The problems could relate to simple data types, arrays, linked lists, as well as stacks and queues. List the advantages and disadvantages of each method of solution to each problem justify the final choice using this information.	• include in the worksheets questions which relate to queues and stacks as well as other data types and structures

Resources

- range of examples of problems requiring different data structures for class discussion
- worksheets giving different data to be stored for students to select and justify the reasons for selection

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.4**	1.4.4 Data management	• explain the difference between backing up and archiving • discuss sensible systems for managing back ups

Classroom Exercises
Describe the processes of backing up (and a sensible system for managing backups) and archiving (to save data which is little used or redundant and would not be restored, but needs to be available for reference). Give the students worksheets which provide descriptions of organisations and data, state whether data needs to be backed up or archived. It may develop further understanding if the marking of the worksheets was oral and interactive.

Resources

- worksheets giving descriptions of organisations and data (need not be real) for students to decide whether archive or backup is the most appropriate

Session Plan Eleven – Hardware

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.5**	1.5.1 Processor components	• describe the function and purpose of the control unit, memory unit and ALU

Classroom Exercises	Notes
Give and then test the basic understanding of the three primary elements of the CPU, covering (briefly) the functions of each element. Reinforce this element orally, via worksheets or using a computer simulation.	• in this section there is no need to go into detail like the fetch-decode-execute cycle

Resources

- worksheets or oral revision/testing

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.5**	1.5.2 Primary and secondary storage	• define the terms RAM and ROM and explain their uses • mention other memory sub categories PROM, EPROM, EAROM, SRAM, DRAM and explain them as sub-types of the two main categories

Classroom Exercises	Notes
Teach the two main categories and their uses. Include volatility and refreshing.	• there is no need for any detailed explanation of the sub categories at this stage in the course

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.5**	1.5.2 Primary and secondary storage	• describe secondary storage in both magnetic and optical media • explain the features of, advantages and disadvantages of each medium • classify examples of data storage as magnetic or optical

Classroom Exercises	Notes
Discuss magnetic and optical storage media. Explain the features of each type along with its advantages and limitations. Discuss speed of access and capacity of each drive type. An enhancement exercise for this would be to discuss the relative merits of each drive type in terms of different access methods. Relate this work to previous work covered on different types of access. Further enhancement could be provided by discussing compression which could be used with these media.	Ensure coverage of: • magnetic Tape • floppy disk (magnetic) • hard disk (magnetic) • CD-ROM (optical) (CD-R) • CD-RW (optical) Mention • zip drives (magnetic) • DVD (optical)

Resources

- worksheets to identify storage media mapped to features, advantages and disadvantages

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.5**	1.5.2 Primary and secondary storage	• describe buffering between primary and secondary storage • describe the purpose of interrupts in the data transfer process relating to buffers

Classroom Exercises
Explain the purpose of buffering, in data transfer between primary and secondary storage. Describe the purpose of interrupts in this process. Enhancement in this section would be to lead a general discussion on interrupts in general and buffering between processor and peripheral devices.

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.5**	1.5.3 Peripheral devices	• describe a range of peripheral devices. For each identify features, advantages and disadvantages • give a range of suitable applications for each device

Classroom Exercises	Notes
Begin by questioning students about input and output devices: • how do you tell the computer what you want to do? • can you think of another way? • how do you know what the computer has done with your information? • how does the computer present information that is not on the screen? Extend the students' ideas to describe the full range of common peripheral devices (including as a minimum: keyboard, mouse, joystick, modem, printer, plotter, barcode reader, MICR, OMR, OCR, scanner, graphics tablet, touch screen, interactive white board, monitor, multimedia data projector, loudspeakers, microphone) giving the features, advantages and disadvantages of each. Set the students an exercise to map these devices to a series of applications. Each mapping must be justified. Mark this exercise orally to promote discussion about the right and wrong answers and in particular the justification for the answers.	Have devices like old floppy discs (which have been opened), CD ROMs, printers etc so that students have visual images of a range of devices.

Resources

- samples of / pictures of different devices and storage media
- worksheet containing pictures of a range of input, output and storage devices; to test the name, type, purpose, and one common application for each picture
- worksheet containing applications and data to be collected. Students to find right devices

Session Plan 12 – Data Transmission and Networks

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.6**	1.6.1 Data Transmission	• describe the characteristics of a Local Area Network (LAN) – particularly sharing resources • describe the characteristics of a Wide Area Network (WAN) • discuss the hardware and software requirements for LANs and WANs to function

Classroom Exercises
Using visual images, describe the characteristics of LAN, particularly in relation to resource sharing – hardware and software. Describe the characteristics of WAN, particularly with increased distance, advantages of resource sharing minimise as distance increases, although not as much in terms of software. Discuss modems and NICs. Describe both the hardware and software required to enable the smooth operation. This may be better done by describing several case studies (including the system that the students are using) and should include some discussion of the dangers from viruses and unauthorised entry.

Resources

- prepared graphical interpretation of WAN and LAN systems (hopefully including the system the students are using)
- detailed case studies of a number of LANs and WANs including hardware and software

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.6**	1.6.4 Networking	• describe for each type of network topology the relative strengths and weaknesses

Classroom Exercises	Notes
For each type of network, use large network diagrams (preferably of systems that the students are familiar with), to help describe the three main network topologies - bus, star and ring. For each type describe its relative strengths and weaknesses. For example: Bus network – lots of traffic down a single spine. Limitations of distance (300m) without need for signal boosting. If problems with the line whole system / spine segment is down. Traffic collision and the potential for monitoring network traffic from another workstation etc. Also advantages – relative cost, easy to install and monitor (single line) This needs to be repeated for each type of topology. Go on to discuss hybridisation of these basic topologies and a basic description of the functional elements of these hybridised networks, like routers, as well as the relative merits of having different network speeds at different points – e.g. Gigabit link between server and router, cable links to increase distance, 100mbit link to frequent users and 10base2 links to infrequent/lower priority users. Discuss the balance between costs of hardware (including NICs) and performance.	

Resources

- prepared large topology maps of networks

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.6**	1.6.1 Data Transmission	• describe the characteristics and uses of Serial and Parallel transmission of data • discuss the use of check sums and parity bits to reduce transmission errors • describe the characteristics of Simplex, Duplex and Half Duplex data transmission • explain bit rate/baud rate

Classroom Exercises	Notes
Describe the characteristics and uses of serial and parallel transmission of data. Use simple 8 bit bytes in the demonstration. In serial each bit one after the other, next not sent until last received and in parallel all 8 bits together on 8 wires. Discuss relative merits in terms of data transmission speed and accuracy. These can be related to peripheral devices. Describe the characteristics and uses of simplex, duplex, half duplex and briefly mention multiplex methods of transmission of data. Good analogies to use are television signals and/or teletext for simplex, telephone or Internet chat for duplex (both people can speak at the same time), and CB radio for half-duplex. Discuss the need for check digits, parity checks and checksums as well as other data checking systems at this point. Include notes on echoing back – to include the need for Duplex or Half –Duplex to allow this to happen. The students should be able to calculate check digits / parity bits and checksums and simulate data transmission and reception to ensure accuracy. Use 8 bit bytes under ASCII with the eighth bit being used as a parity bit (both odd and even should be understood). Pre prepared worksheets containing messages to calculate these values and another bank of questions to decrypt to check if the message has been transmitted correctly would be useful at this point. Describe the term BIT rate as a transmission speed of 1 bit / second and the term baud which is used to measure the number of bits / second. 1 baud = a transmission speed of 1 bit per second. Discuss (using the earlier work on ASCII etc.) to transmission speeds for text and graphics and relate this (using the Internet as the background) to the need for small file sizes, and particularly file compression.	• ensure that the term Baud is understood • it may be worth briefly discussing the need for data/file compression at the same time as this topic

Resources

- two types of worksheet:

    1. calculate parity bits, checksums for a variety of data to be transmitted
    2. check received data using odd parity/even parity, check digits and checksums to see if data has been received correctly

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.6**	1.6.2 Circuit switching and packet switching	• describe packet switching and circuit switching

Classroom Exercises

Explain using a large topological map of a WAN (preferably related to the Internet) how different packets of data can be routed in different ways to the same destination.

Describe packet switching – explain the process of segmenting the message / data to be transmitted into several smaller packets. Each packet is labelled with its destination and the number of the packet. Each is despatched and many may go via different routes. The original message is reassembled in the correct order at the destination.

Describe circuit switching – a route is reserved from source to destination and the entire message is sent in order and therefore does not need to be reassembled at the destination.

Resources

- network diagram showing a number of nodes interlinked

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.6**	1.6.3 Protocols	• explain network protocols • explain the need for both machines involved in the data transmission / reception to be configured to use the same protocols • describe the layering of interfaces

Classroom Exercises	Notes
Explain protocols as the rules that govern the transmission and reception of data. Briefly mention commonly used protocols like TCP/IP and HTTP and explain their function. Briefly explain the need for both machines involved in the data transmission / reception to be configured to use the same protocols. Describe the layering of interfaces and the need for both machines to be using the same methods of error checking and correction, how routing will take place, how data flow will be controlled and data synchronisation (so that the data does not become mixed up – especially in the case of switch streaming) etc.	Candidates do not need detailed knowledge of specific protocols.

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.6**	1.6.4 Networking	• explain the advantages and disadvantages of networking

Classroom Exercises

Review work done on networks and lead a classroom discussion about general advantages and disadvantages of networking, e.g. shared resources (hardware and software), communications, cost etc. Make notes on the board and from these the students compile their own set of notes.

Students are then presented with worksheets where they are given two systems analysis examples for new computer systems where they have to list and justify the relative advantages and disadvantages of networking the computer system. Try to include one system where a WAN would (arguably) be beneficial and one for a LAN. After the students have worked the examples brainstorm the answers collectively to share ideas and promote discussion on the underlying issues.

Resources

- worksheet giving two different scenarios of systems analysis to analyse whether networking would/would not be beneficial

Session Plan 13 – Systems Life Cycle

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.7**	1.7.1 1.7.2 1.7.3 1.7.4 1.7.5 1.7.6 1.7.7 1.7.8 1.7.9	• introduce the nine stages of the system life cycle • explain the system life cycle as an iterative process

Classroom Exercises	Notes
Introduce the nine stages of the system life cycle: Identification Feasibility study Information Collection Analysis Design Development and Testing Implementation Maintenance Obsolescence Explain the system life cycle as an **iterative** process: it should be seen as a continually developing process. There will be a need to reconsider and review all previous stages as each subsequent stage is completed. Give scenarios where previous stages would need to be revisited and specifications changed.	

Resources

- list of scenarios which would require iterative development

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.7**	1.7.1 Identification of problem	Describe what is meant by identifying the problem.

Classroom Exercises	Notes
Provide a number of examples describing situations which need a computer system to be implemented. The examples should be of increasing complexity. Ask students to suggest what the problem is and then compare answers. Explain the importance of defining the problem clearly and accurately. The importance of having the aims of the system being agreed by all those involved at this stage must be stressed. The initial discussions between the systems analyst and the 'client' organisation must ensure that the analyst fully understands the nature of the problem and the business of the client. There must be discussion between all the interested parties, and then a list of objectives is written up. This list of objectives, if they are all solved, will be the solution to the problem. All the people involved agree to the list of the objectives, or they are revamped until all can agree. The completion of these objectives is the success indicator for the project.	Make sure that students understand that the life cycle can be terminated at any point.

Resources

- examples of situations where students have to suggest what the problem is

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.7**	1.7.2 Feasibility Study	Discuss the process, nature and purpose of a feasibility study

Classroom Exercises	Notes
Discuss what a feasibility study is, relating it to a common real-life example. (Is it feasible to offer all students a free coffee each day? Why not?) Ensure that the students understand the meaning of the word feasible. Describe the process of the feasibility study. It should include a decision on how valuable a computerised solution is to meeting the objectives identified within the definition of the problem. The analyst will report on what is possible and sensible given the objectives. If the feasibility study shows that the solution is viable after these stages then the analyst moves on to consider the collection of information. Using the examples from the previous exercise, ask students to carry out feasibility studies.	The following elements should be included in every feasibility study: • is the solution technically possible? • is the solution economic to produce? • is the solution economic to run? • what is the effect on employment? • what will be the skill requirements of the workforce? • what effect will there be on the customer? • will the solution increase profitability?

Resources

- examples (from previously) of situations where students carry out feasibility studies

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.7**	1.7.3 Information collection	Discuss the information a system needs and how to collect it.

Classroom Exercises	Notes
Describe and explain the importance of determining the information requirements of a system. Include within this the methods of fact finding using questionnaires (reminding students about good form design), individual interviews and group meetings and ask students to suggest the advantages and disadvantages of each method. Discuss the sources of information at this stage, for example: • employees of companies can often identify possible changes in working practice / method which could enhance the business but have not had a way to express these views • observation of the existing systems at work (taking into account the changes in behaviour / approach that some workers may demonstrate when being observed.) Use prepared worksheets with details of the information required and questions to select the best methods of collecting data giving the students a number of simplistic scenarios with pre determined objectives. The resulting discussion / marking / brainstorming session should create discussion to enhance the students' analytical thought process in this area.	Point out the fact that observation of the existing documentation and other paperwork should also be undertaken to fully familiarise the analyst with the existing system.

Resources

- worksheets with details of the information required and questions to select the best methods of collecting data giving the students a number of simplistic scenarios with pre determined objectives

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.7**	1.7.4 Analysis of a problem, based upon information collected, including producing a requirements specification 1.7.5 Design of a system to fit requirements	• describe the processes of determining the type and amount of data to be stored • Introduce JSP and data flow diagrams (system flowcharts) • describe how to design the input, output and processing elements of the system • describe how modularity (often taken from the JSP) assists in the program design

Classroom Exercises	Notes
Ask for the students' ideas about how to determine the type of data to be stored in a given system. How much data is needed? Make them aware of the fact that the decisions about the nature and amount of data will influence the software and hardware requirements of the system. Issues should include as a minimum: • types of data to be held • form of data storage - to calculate overall storage space for each set of data • number of sets of data – to calculate overall storage space, devices and the effect on data structures that this may have • types of access to the data • frequency of update and access to the data The details of data are often structured using data flow diagrams (system flowcharts) and Jackson diagrams. This would be a good point to introduce Jackson Structured Programming, setting a number of small exercises to be solved using JSP. This will also help to reinforce the concept of program modularity. Design specifications (other constraints) would already have been placed on the design. All the elements listed above would be agreed between the analyst and clients before implementation. Explain how the design specifications include the input and output requirements (taken from the Identification section), and the processing requirements. Describe how the input and output requirements can be refined. Describe how the processing of data can be structured into modules and each element tested.	JSP exercises are designed to give students a feel for the process, not an in depth study. Input requirements can be refined using these elements: • what data is required? – This is taken from the identification and data collection stages. What format should this be in – e.g. Text, graphical etc. Does the data exist or does it have to be captured / collected first • the hardware that is available and/or required? Is data entry to be automated / manual? • the experience of the operator • the design of the user interface Refinements to output requirements could include: screen design, what information can be output automatically, (form letters, email messages etc). ways to attract the operator's attention to elements of the process at certain times by user interface enhancements – e.g. colour change, flashing etc.

Resources

- exercises in simple Jackson Structured Programming. These should not relate to large or complex tasks

Session Plan 14 – Systems Life Cycle

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.7**	1.7.6 Development and testing of a system	• describe modular design using constraints of input, output and process for each module and testing of the modules • detail the on going evaluation process, checking whether this matches the requirement specifications • explain the types of documentation (including content and importance) that it is necessary to produce alongside the development of the system

Classroom Exercises	Notes
Describe how the processing of data can be structured into modules and each element tested. Explain the modular requirements for on entry, on exit and process. Describe how at each stage of the system life cycle constant evaluation is needed – but especially at this stage. Does the system, as it is developing, match all the criteria in the identification of the problem? Explain the need for documentation at each stage of the system life cycle. It must explain: • how the system has been produced • how it should be used • how it can be maintained It may be worth introducing all the documentation elements at this stage – although much greater depth will be required for the assignment later in the course.	The following documentation should be included: • requirements specification • design specification • program specifications • technical documentation • user documentation

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.7**	1.7.7 Implementation of system	• review the work on system testing • describe the initial three stages of implementation • describe and discuss the different methods of implementation

Classroom Exercises	Notes
Review the work from session eleven on testing, discussing test plans, alpha and beta testing in the context of system implementation Introduce the idea of a planned implementation rather than an ad-hoc introduction. Introduce the initial three stages of implementation: • ensuring that the correct hardware is available • ensuring staff are trained in the management and/or use of the new system • initially structuring and entering the system data, either manually or by downloading them from the original system Describe and discuss the different methods of implementation. Pay attention to the advantages and disadvantages of each method. Give the students some pre- determined scenarios which require them to take decisions on how to implement them. These could be from prepared worksheets. (Try to include critical examples like an air traffic control system, or replacing an existing traffic light system in a major city.)	Methods of implementation should include: • parallel running • pilot running • big bang • phasing

Resources

- worksheets with suggested scenarios for different types of implementation

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.7**	1.7.8 Maintenance of system 1.7.9 Obsolescence	Discuss: • the possible reasons for maintenance of a system • system review and reassessment • planned / unplanned obsolescence

Classroom Exercises

Question the students: once a system is in place, why would you need to do further work (maintenance) on it? What reasons can you think of? Drawing on the students' responses, discuss the possible reasons for maintenance of a system which may include errors in the software, changes in legislation (which might include changes in tax rates etc), the original specifications are changed, hardware may be upgraded/changed. This should reinforce the need for maintainer documentation

Discuss the need for constant system review and reassessment, particularly related to the limited life span of hardware and software platforms and the current trend for upgradability.

Discuss planned and unplanned obsolescence. This can be done by offering a historical scenario, such as a system is running on a particular platform and a newer faster platform appears, and the decision making processes that would follow this scenario.

Resources

- possible scenarios to provoke discussion of unplanned obsolescence

Session Plan 15 – Applications Software

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.8**	1.8.1 Custom written software versus off-the-shelf software packages	Discuss the relative advantages and disadvantages of using off-the-shelf and bespoke software packages.

Classroom Exercises

Discuss the relative advantages and disadvantages of using off-the-shelf and bespoke software packages. Some advantages of off-the-shelf packages include:

- it is immediately available, bespoke software takes time to write
- it will have many users who share the development costs, making it a cheaper alternative
- it will have been more thoroughly tested (due to the number and variety of users)
- it is more likely to be compatible with other applications packages
- it is more likely that there are well established training courses in the software

Some advantages of bespoke packages (custom written software) include:

- software will be tailored to the exact needs of the user
- perhaps no off-the-shelf software fulfils the system requirements
- there is a potential to work with the developers to expand the marketplace for the new software
- not paying for areas/routines that are not going to be used

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.8**	1.8.2 Application Areas	Describe the features of common business applications.

Classroom Exercises	Notes
For each of the following areas discuss the features of applications packages designed for the function: • stock control • order processing • payroll • process control • POS systems • marketing • CAD • CAM One approach to this is to use concrete examples from local business (if they will provide you with details). The students will relate much better to real examples.	Include discussion of the purpose of the package, the inputs and data validation/verification methods, processing and output. Relate to work on bespoke and off-the-shelf packages and discuss how, in some cases, off the shelf packages can be adapted to meet the needs of a given business. **Note**: Some companies may be sensitive about releasing any details about their operations.

Resources

- details of local companies who operate the areas noted (bulleted list), explaining the function of the software and the part that it plays within the organisation. Exact details do not need to be obtained but general information should suffice

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.8**	1.8.3 Applications Software	• discuss generic applications in general • describe common generic applications • discuss when it is not appropriate to use a generic applications package

Classroom Exercises	Notes
Begin with a questioning session. What is a word processor? How many word processors can you name? What can all word processors do? Now repeat with spreadsheets and databases and begin to build up a picture of what generic software is. Define the word generic. Review the features and uses of further generic applications packages. After the initial introduction of these features give the students an exercise with a series of real life scenarios where the students must determine the correct application to fit the scenario. The marking of these elements would be better as an interactive session so that the reasoning behind each solution can be explained. To extend this discuss the areas where these generic applications packages would not be appropriate. Again an interactive session with students suggesting scenarios and discussion of the requirements would be beneficial – for example control systems, robotic production lines and elements of stock control systems. Discuss the purpose and impact of these generic applications packages – e.g. mail merge has developed from word processing, desktop publishing has led to an increased number of publications because there is no requirement for typesetting which was expensive, highly skilled and therefore expensive.	Cover these areas: • Word processing • Spreadsheets • Databases • Desktop Publishers • Presentation software • Graphics packages Relate the discussion to work on off-the-shelf and bespoke packages. Discuss how, for example, a spreadsheet can be programmed to provide a bespoke package. Extend the discussion of the impact of generic applications like mail-merge on society – e.g. junk mail. This will be covered in more detail in session 18.

Resources

- worksheet giving a number of real life scenarios against which the students can map appropriate applications packages

Session Plan 16 - Handling of data in Information Systems

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.9**	1.9.1 Data Capture, preparation and data input	Discuss the meaning of data capture (to include images) using a range of both manual and automatic examples. Explain how the data captured can be entered onto a computer system.

Classroom Exercises	Notes
Present a number of scenarios which require data capture, including examples of both automatic and manual data capture (for example: filling in a form by hand at a bank, the use of a bar code reader at the supermarket, taking a photograph). Ask the students to think about what data is being captured, how it is being captured and how it could be input to a computer system. Describe manual methods of data collection and ask students to suggest further examples. Focus on data capture forms and in particular form design. Introduce the students to the features of good form design, and the collection of the right sort of data – e.g. use of date of birth instead of age. Explain the process of transferring the data manually from the form/s into the system, primarily through keyboard entry. Discuss automatic methods of data capture and provide a handout describing different methods and devices used to capture data automatically, with examples. Include the advantages and disadvantages of each for specific purposes. Describe the capture of images using scanners, video capture and digital cameras, explaining how each system works and situations where one system of image capture is better than others. Discussion of these areas will reinforce learning.	Include: • Sensors • Data loggers • Speech recognition • Touchscreen • Barcode reading • OMR • OCR • Magnetic stripe cards Briefly explain the terms analogue and digital relating to every day examples such as clocks and watches. Explain why we need to convert data from analogue to digital using an ADC to take readings like barometer pressure and convert them into digital data.

Resources

- worksheets giving a variety of scenarios to which different data capture methods can be applied
- handouts – notes

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.9**	1.9.2 Validation and verification of data	Discuss the need for the accurate input of data and the ways in which we can check that the data is correct.

Classroom Exercises	Notes
Make students aware of the fact that data can be checked both automatically and manually. Ask them to suggest the limitations of both methods. Will a computer **know** there is a mistake if a date of birth is typed in as 16/12/85? How about 16/13/85? Describe the meaning of the term valid and emphasise the fact that a computer can only check for **valid** data. Look at checks for existence, range, character, format, length and check digit (in the case of bar codes etc) as automated on data entry. Discuss examples of data validation students may have encountered, for example error messages. Demonstrate how to create a simple range check in Excel or another spreadsheet. Discuss what verification means. Describe verification of data as manual checking that the data has been typed in correctly, sometimes visually but more often by double data entry. Exercises should be given to invent algorithms which check for each validation category in the language of their choice (or in flow diagram form if preferred). As extension work algorithms could even be developed to perform verification checks using double data entry on a byte by byte basis.	• developing validation algorithms gives far better retention to students but do not over-emphasise • ensure that students do not confuse validation with parity checking or data transmission/receipt error checking

Resources

- worksheet/s containing exercises to develop algorithms to validate data

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.9**	1.9.3 Outputs from a system	• describe the variety and scope of output formats from a system • discuss using examples and consider relevance of the data presented to the task • discuss the timing of animations, video, sound and presentations and their critical effect upon the overall impression on the audience

Classroom Exercises	Notes
Revise output and output devices. Ensure that students are aware of the fact that data can be output in a variety of ways and relate each output format to a range of applications in the real world, hopefully to local companies or those that the students are familiar with. Students should begin to appreciate: • the benefits of using a range of output formats • the relationship between the data and the way it is output For each format ensure that the discussion covers: • the advantages and disadvantages • the selection of output format to match the target audience Using prepared worksheets giving a number of scenarios where the students are told the target audience and the nature of the information to be given, working in groups discuss and decide upon the most appropriate formats for the output. Each group should present its ideas and the interactive discussion/marking of these ideas should develop a sound understanding of these concepts.	Cover these areas: • images • animations • video • interactive presentations • graphs • reports • sound Discuss timing of animations, video, sound and presentations and their critical effect to the impression on the audience.

Resources

- worksheets containing details of target audience and information to be presented for a number of scenarios

Session Plan 17 – Designing the User Interface

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.10**	1.10.1 Interface Design	• discuss interface design • set students an exercise to design interfaces

Classroom Exercises	Notes
Provide screenshots or on-screen examples of a range of interfaces and ask students to evaluate them in terms of: • target audience for the interface • type of data to be collected, information to be given • circumstances/conditions that user interface will be used in • how effective the communication is • user enjoyment (in some circumstances) Discuss the importance of good interface design. Discuss the Human Computer Interface (HCI) design issues, asking the students to suggest features to be considered and why they are relevant. It may be beneficial to expand this discussion to include short term and long term memory as well as the users' visual perception of the information. Discuss the styles of interface and their relevance to application design. Set an exercise to design three user interfaces for a web page to show information from given design briefs. Make sure that the nature of the data to be conveyed is very different (include text, graphic, sound) and the target audiences very different (children, all adults, university professors). Ensure that one of the contexts includes a form to collect data. The user interfaces are to be designed on paper. Debrief by discussing the strengths and weaknesses from the designs suggested. Merge the best points of all to create good 'model answers'. The process of creating these model answers is more important than the end result.	For styles cover these areas: • forms • menus • command line • natural language • speech • direct manipulation For HCI include these areas: • colour • layout • content

Resources

- screenshots or on-screen examples of a range of interfaces
- worksheets containing design briefs with a variety of target audiences and forms and styles of information to be input and output

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.10**	1.10.2 Criteria for selecting appropriate hardware	Discuss the ways in which users are able to communicate with the user interface. Progress to the hardware that provides the interface and the criteria used in the selection of hardware.

Classroom Exercises	Notes
Discuss the importance of selecting the correct peripheral devices for both input and output. Ensure that for each peripheral considered there are suitable applications and develop the students' awareness of selecting and justifying their choice of device. Discuss the required characteristics of the user interface taking account of: • information to be used • type of user • physical location • current technology Discuss the potential problem of speed mismatch between the user, peripheral device and processor. Set an exercise to select appropriate input/output devices for a user interface from given design briefs. Make sure that the nature of the data to be conveyed is very different (include text, graphic, sound) and the target audiences very different (children, all adults, university professors). Ensure that some of the contexts include a form to collect data. The user interfaces are to be designed on paper.	Hardware - include as a minimum: • Keyboard • Mouse • Joystick • Modem • Printer • Plotter • Barcode reader • MICR • OMR • OCR • Scanner • Graphics Tablet • Touch screen • Active white board • Monitor • Multimedia projector • Loudspeakers • Microphone

Resources

- worksheets containing design briefs with a variety of target audiences and forms and styles of information to be input and output

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.11**	1.11.1 Passive versus interactive systems	Compare and contrast passive and interactive information systems through class discussion and questioning.

Classroom Exercises

Students name five passive and five interactive information systems (no prior help given).

Discuss their attempted answers, focusing on the characteristics of passive information systems. Examples can be found on CD ROM (e.g. Encyclopaedia, teletext) and on the Internet – be careful not to select sites with email responses or forms. Other examples could be searching a library system for a book (as a customer, not a librarian).

Now focus on examples of interactive information systems. Use Internet examples that contain email responses, and/or forms, use of a database where the user can edit the data (students' records). POS terminal with stock control etc.

Close with a comparison of passive and interactive systems – students take notes.

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.11**	1.11.2 Characteristics and uses of management information systems	• describe MIS as a system which allows managers to access and analyse data • explain briefly strategic management

Classroom Exercises

Describe MIS as a system which allows managers to access and analyse data. Briefly explain the difference between strategic management and condition driven management.

Give the students an exercise containing a list of jobs/problems/scenarios which could occur in the running of a business. The students have to decide which tasks are strategic management and which are condition driven management and detail which elements of a MIS could be used (and how) to help them solve the job/problem/scenario.

Verbal debrief on the above exercise.

Resources

- a list of tasks on a worksheet for the students to categorise as strategic management or condition driven management

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.11**	1.11.3 Batch processing and rapid response applications	Discuss batch mode processing and rapid response processing.
Classroom Exercises		
Describe batch mode processing and rapid response processing – do not use more than one example for each. Give the students an exercise to list six batch processing tasks and six rapid response processing tasks. Discuss the students' responses to the task.		

Session Plan 18 – Implications of Computer Use

Assessment Objectives	Performance Criteria	Classroom Ideas
• **1.12**	1.12.1 Economic Implications 1.12.2 Social Implications 1.12.3 Legal Implications 1.12.4 Ethical Implications	Find out what the students know and think about the use of computers in society and guide their discussion to cover the economic, social legal and ethical implications of the increased use of computers. Help students make notes on these areas and ensure that they have covered all aspects of the topics.

Classroom Exercises	Notes
Begin with a classroom discussion about how computers are used in society, in the workplace, in the home and in education. Ask the students to suggest ways in which the use of computers may have changed aspects of society and to identify which of these could be seen as problems. Why are they problems? Most students will probably have an idea about the developments in computing in recent years, and ask them to identify what these developments are and how each development is related to the effects of computers in society. For example, the use of the Internet is now widespread in schools owing to the developments in personal computing and networking technology; access to the Internet could have social and ethical implications for children at school especially if they find undesirable sites such as pornography. Hand out prepared worksheets outlining the major developments in computing history and ask the students to work in groups to think about how each development affected society. Guide a class discussion of responses to include major points about health and safety issues, data protection and the need for confidentiality of data. Provide resource materials such as books and suitable web sites and an outline structure for a set of notes on the topic. Ensure that the notes cover: • discuss the trends in computer use and their effects on society • explain the changes to society brought about by the introduction and use of computer systems • explain the need for data protection legislation and the current legislation in your country • discuss the social and ethical implications of access to information whose value is controversial (e.g. human rights, paedophilia and the potential for terrorism) • discuss Health and Safety implications (and where it exists) the current legislation • explain the need to protect confidentiality of Data and how this can be addressed	Students' notes should include: • communication e.g. mobile telephones and email • changing work patterns, e.g. home-working, loss of traditional jobs and the creation of new jobs • increasing quantities of personal data being held on different computer databases and the potential for data profiles to be built • the fact that information on the Internet is largely uncensored and can be posted by anyone • health and safety issues related to seating, posture, RSI, lighting, eye strain, stress and periods of inactivity (DVT)