SEC SYLLABUS (2006-7)

COMPUTER STUDIES SEC 09

SYLLABUS

Computer Studies SEC 09 (Not available in September)

Syllabus Paper I (2 hrs) + Paper II (2 hrs) + Coursework

This syllabus assumes that schools and other educational establishments preparing candidates for this examination will provide suitable facilities for practical computer experience in:

- Using an operating system with a graphical user interface;
- Using a variety of standard application packages – as a minimum a word processor, a database, a spreadsheet and a graphics package;
- Use a variety of standard Internet client packages – as a minimum a Web Browser and an e-mail package;
- Writing, debugging and testing programs in a high level language.

The syllabus is designed to be covered over a period of 3 scholastic years typically with 4 lessons of 45 minutes per week. The total hours covered is 240 hours. The practical part could take up to 100 hours of the whole course.

Aims
This syllabus aims to:

1. stimulate and foster an interest in the use of computers;
2. develop practical skills in the use of computers;
3. develop skills in creatively applying information processing technology to problem solving;
4. provide a broad view of the range and variety of computer systems and applications;
5. develop the ability to communicate and interpret information and concepts relevant to computing;
6. introduce students to the fundamental concepts of computer science;
7. serves as a basis for further studies in Intermediate and Advanced level in Computing or Information Technology.

Structure of the syllabus
The syllabus is organised into 5 main parts, as follows:

8. Computer Applications
9. Hardware
10. Computer Systems
11. Algorithmic Problem Solving and Programming
12. Information and Communications Technology in Society

This structure is meant to serve as a framework for teaching the subject, in that topics have been arranged to reflect a logical progression. It is strongly suggested that teachers follow this ordering of topics.

Core and extension topics: The syllabus is further divided into a core component and a set of extension topics. The extension topics are marked with an (and are printed in *italicised* text in the syllabus. Questions about extension topics will be limited to paper IIA.

Supplementary notes: The supplementary notes are included in the right column next to the syllabus and they contain guidelines for teachers and elaboration of the syllabus content. These notes are to be considered an integral part of the syllabus.

Course work (Appendix 1): Appendix 1 gives details about the coursework assessment criteria which will guide the Markers' Panel during the moderation.

ASSESSMENT OBJECTIVES

The examination tests the candidates' ability to:

- recall and understand information and concepts;
- analyse simple problem situations and evaluate feasible computer solutions;
- apply their knowledge of computing to solve simple problems in both familiar and unfamiliar situations;
- keep abreast with current developments, trends and issues related to everyday use of computers.

SCHEME OF ASSESSMENT

The examination will consist of two written papers of 2 hours' duration each, and an assessment of practical work. The maximum mark that may be obtained is 200. The questions will be set in English and must be answered in English.

Paper I (2 hours, maximum mark 85)

This paper will consist of between 10 and 15 short, compulsory questions covering the **core** syllabus. All questions are to be answered in the space provided on the examination paper itself. **All** candidates registered for the examination must take this paper.

Paper II (2 hours, maximum mark 85)

This paper consists of eight longer questions, of which candidates must choose five. There will be two versions of this paper - option A and option B. Candidates are required to indicate on the registration form which option they wish to sit for.

Paper IIA

Questions in this paper will be more difficult than those in Paper I, and will cover BOTH the core topics and the extensions. Candidates opting for this paper may qualify for grades 1, 2, 3, 4 or 5. The results of candidates who do not qualify for at least a grade 5 shall remain Unclassified (U).

Paper IIB

Questions in this paper will be easier than those in Paper I, and will cover only the core topics. Candidates opting for this paper may qualify for grades 4, 5, 6 or 7. The results of candidates who do not qualify for at least a grade 7 shall remain Unclassified (U).

Flowchart templates may be used in both written papers. Calculators may not be used in either paper.

Coursework (maximum mark 30)

The four exercises listed in **Appendix 1 - Coursework** should be completed by the candidates during their normal course of study under the supervision of their tutor or tutors. Candidates should note that:

- ALL coursework MUST be word processed. Marks for word processing are allocated in a separate marking section to the other four exercises
- Excessive length will NOT contribute towards the final mark.
- Tutors should use the five Marking Schemes included in Appendix 1 when marking the exercises and the overall marks obtained for each exercise should be entered in the appropriate form, which also includes a marking scheme (see Appendix 1). For each candidate, schools or tutors should keep a folder containing:
  - a copy of the five marking schemes, showing the mark awarded by the tutor for each section of the exercise;
  - when providing Coursework marks for Matsec, tutors are requested to quote the mark out of 30 (i.e. 150 divided by 5);
  - the candidates exercises.

No diskettes are to be included in the coursework folder. The Marker's Panel will be responsible for moderating the tutor-assigned marks and candidates may be called for an interview relating to their coursework exercises.

Private Candidates

(a) Candidates who are re-sitting the subject may carry forward the coursework mark from a previous session.
(b) Candidates who have never studied the subject at school but have covered the coursework privately will be expected to present the four exercises to the MATSEC Board when instructed to do so by the board. Candidates may be called for an interview about their work.

GRADE DESCRIPTIONS

The following grade descriptors give a general indication of the level of attainment reached by a candidate for SEC Computer Studies. The descriptors are related to syllabus content and not designed to define that content.

GRADE 7

Candidates:

- show a basic knowledge and understanding of the wide-range use of computers today in information processing
- should show basic ability in using common office application software along with communications software
- show basic knowledge of computer hardware and peripherals.
- show good knowledge of how the various components function and communicate together as a complete system
- should have basic ability in using one specific operating system and should be aware of the existence of other operating systems and networks
- should be aware of the main stages in systems analysis
- should appreciate the widespread use of ICT worldwide
- are able to solve simple linear algorithms
- show familiarity with the simple constructs of the programming language studied and interpret simple programs
- should have a basic understanding of the practical problems involved when using computers

GRADE 5

Candidates:

- show average knowledge and understanding of the wide- range use of computers today in information processing
- should show an average ability in using common office application software along with communications software
- show average knowledge of computer hardware and peripherals
- show average knowledge of how the various components function and communicate together as a complete system
- should have average ability in using one specific operating system and should be aware of the existence of other operating systems, networks and their characteristics
- should have a wider understanding of the main stages in systems analysis
- should appreciate the widespread use of ICT and its effects on the wider world
- are able to solve more complex algorithms including decisions
- show a wider knowledge of simple constructs of the programming language studied including the coding, interpretation and testing of simple programs
- should have an understanding of the practical problems involved when using computers.

GRADE 1

Candidates:

- show a good knowledge and understanding of the wide-range use of computers today in information processing
- should show versatility in using common office application software along with communications software
- show more detailed knowledge of computer hardware and peripherals
- show a deeper knowledge of how the various components function and communicate together as a complete system
- show greater ability in using one specific operating system and should be aware of the existence of other operating systems, networks and their characteristics
- should have a deeper understanding of the main stages in systems analysis and be able to apply them in system development
- should be able to differentiate amongst and appreciate the widespread use of ICT applications and their effects on the wider world
- show mastery in solving complex algorithms including simple low level language problems
- show a deeper knowledge of the constructs of the programming language studied including the coding, interpretation and testing of more complex programs
- should have a good understanding of the practical problems involved when using computers.

SYLLABUS

PART 1
COMPUTER APPLICATIONS

SYLLABUS	SUPPLEMENTARY NOTES
OBJECTIVES The candidate should be aware of the widespread use of computers today in processing information and should have first-hand experience of the use of common office application software.	The primary aim of PART 1 is to introduce the computer system from the **user's** point of view, using **practical** examples. This is intended to: 1. motivate candidates, showing them the power and potential of computer systems, and 2. provide candidates with a first-hand knowledge of various topics to be discussed in greater detail later in the course. The teacher should be able to draw on the practical experience gained by candidates in this part to illustrate points arising in the more theoretical parts of the course.
1.1 THE COMPUTER SYSTEM The computer system as an information processing machine. Its tasks of handling information: inputting, processing, outputting, storing, retrieving, sending and receiving information.	The teacher is to introduce this section with a demonstration application encompassing word processing, spreadsheet and database, as explained above. Students can then modify an existing spreadsheet and watch the effect on screen, perform a mail-merge using the word processor,

	etc. It is required that the teacher design a carefully thought-out demonstration application from everyday life. The application should encompass the most widely-used packages nowadays - spreadsheet, database and word processor. Moreover, it is imperative that the demonstration should emphasise the **integrated** use of these packages, and not merely present each one in isolation.
1.2 PREPARATION OF DATA FOR INPUTTING INTO THE COMPUTER SYSTEM Data capture forms (very simple), preparation and transcription of data with their related errors; solution through data verification and validation; check digits; range check.	For example, a stock control application can be used to show the connection between analysing the system, preparation of data, using a database to store records of items, using a spreadsheet for calculations of profit or loss and a word processor for mail merge. Knowledge and structure of one application (e.g. stock control, payroll, lending library, school administration, utility accounting, hospital, billing) is required. Security of files, privacy of information in files and file generations (grandfather, father, son) are covered in Part 5, but should be briefly mentioned here in the context of the demonstration example.
1.3 SERIAL AND DIRECT METHODS OF ACCESS Serial and direct access and their suitability of use for certain applications (e.g. serial for payrolls, direct access for airline booking reservations).	Teachers should emphasise that serial and direct are access MODES - some devices are capable of supporting both access modes (e.g. disk), while others can only support serial access (e.g. tape).
1.4 BASIC FEATURES OF AN OPERATING SYSTEM Running several applications concurrently in different windows, easy to use graphics interface, use of clipboard to exchange text and graphics data between applications in particular the integrated use of spreadsheets, word processing and data bases. managing of files: copying, deleting, renaming, creating of folders/directories.	In this section the operating system is considered from the end-user's point of view. A more technically detailed treatment is given under section 3.5.
1.5 COMMON APPLICATION SOFTWARE Classical commercial packages – word processor, database management system, and spreadsheet. Other popular software packages – utilities, anti virus software, graphics. Basic skills in using typical software packages mentioned above. Use of applications software under a windowing	

environment.

The ability to suggest the most suitable software for use in specific environments.

The ability to compare and contrast different packages.

1.6 THE SPREADSHEET

Its use to process information. Only simple understanding required of, e.g., storage of data in cells as a label, value or formula; simple calculations; copying and moving cells; data graphing; printing.

One demonstration of graphs and one example from profit analysis, costing, budgeting, income-tax calculation or projections is recommended.

1.7 THE WORD PROCESSOR

Its advantages and ease of use to produce a simple document such as a letter. Its main features such as fast, easy entry and editing of text, word wrap, margin justifications, centring of text, underlining, indentations, page layout, blocks, find, find/replace, spell check, mail merge.

Candidates should be given a simple example document to be word processed and a simple model to be worked out using a spreadsheet. Similarly, candidates should be introduced to DTP and a drawing package through practical use of these application packages. Examination questions will concentrate on the features commonly supported by these packages, how they differ from each other (e.g. word processor vs. DTP), and their suitability for a particular application.

Main desktop publishing features of Word Processing such as automatic table of contents and index creation, multi-column documents, tables, frames, embedded graphic objects, etc. At least one demonstration of the use of these features should be carried out, giving candidates themselves a chance to create more sophisticated documents.

At least one demonstration of the use of these features should be carried out, giving candidates themselves a chance to create more sophisticated documents.

1.8 DATA BASES

DATA AND INFORMATION

The organization of data in files: files, records (fixed length and variable length records), fields, items, key field.

The teacher is to use a pre-prepared database for demonstration purposes. It is suggested that this database includes records, fields, field types, field overflow, key fields, etc. and is to be made relevant to students' environment.

FILE HANDLING AND MAINTENANCE

Updating of data files, inserting new records, deleting unwanted records, and changing items (fields).

File Reports: retrieval of records to view on computer system or print. Importance of speed of response and file structure (serial or direct access comparisons illustrated by an application). Selecting records under certain conditions (e. g. all customers in a certain

Practical work in this section should include making modifications to an existing database (the one used in the demonstration application), and creating a very simple database (e.g., an address book).

Students are expected to create a simple flat file with file specifications and using a table, queries, form view and report generation.

area). Display or print chosen fields from selected records or from all records.

Sorting a file according to some criteria, e.g. alphabetically by name or by town.

Master files and transactions files - their everyday use.

1.9 GRAPHICS PACKAGES

Basic tools to create a simple graphic and its inclusion into text documents.

It is suggested that students be encouraged to modify clipart pictures and/or create their own using a graphics program before transfer to a text document.

1.10 COMMUNICATIONS TOOLS

THE WEB BROWSER

Simple use of a web browser to access web sites and for searching using a popular search engine. Organising sites in folders. Navigating among sites using the web browser.

E-MAIL

Using an e-mail program to send and receive messages. Saving, printing and deleting a message. Advantages and disadvantages of e-mail compared to the postal system.

CREATING WEB PAGES

Using a web authoring program to edit/create a simple web page that includes text, graphics and buttons for linking to other sites.

No working knowledge of HTML languages is required but viewing of the source code is advisable. Web authoring features included in Office applications may also be used.

PART 2
HARDWARE

SYLLABUS	SUPPLEMENTARY NOTES
OBJECTIVES The candidate should be familiar with the main hardware functional units of a computer system and how these fit together and communicate to form a complete system.	Because this part of the syllabus tends to be somewhat theoretical, special effort should be made by the teacher to present this material in as concrete a way as possible. Wherever possible, students should be shown the hardware units discussed in this section, and preferably be allowed to use/handle them. Other instructional aids required include sample output from the various output devices discussed, and samples of media (diskette, printer ribbon, inkjet head, etc.). Many

excellent videos demonstrating the internal workings of a digital computer in an easy-to-understand and appealing way are currently available.

The features of a computer are frequently mentioned prominently in advertisements and brochures (which are typically intended for the average layman), and for this reason it is felt that candidates should be sufficiently well-informed to make sense of these features.

2.1 COMPONENTS OF A COMPUTER SYSTEM

Main components of a computer system. CPU, I/O subsystem, main and backing store. Flow of data and control between the main units using buses.

An overall view of a computer system should be presented, with each unit as a black box. The following sections will fill in the details.

2.2 COMPUTER LOGIC

Distinction between analogue and discrete processes and quantities. Conversion of analogue quantities to digital form. Using sampling techniques, use of 2-state electronic devices (logic 0 and logic 1) for reliability.

Units of storage - bits, bytes (=8 bits), kilobytes (= 1024 bytes), megabytes (= 1024 kilobytes) and gigabytes (1GB = 1024 megabytes).

The distinction between analogue (continuous) and discrete quantities and processes should be explained carefully. Some textbooks give examples which are more confusing then enlightening – these should be avoided. The main objective of this section is to demonstrate how different data types can be represented using just 2 symbols (0,1), and how data thus represented can be symbolically manipulated.

LOGIC CIRCUITS

Truth values. OR, AND (2-input only) and NOT gates and their truth tables. Determining the output of a logic circuit containing the three mentioned gates for given inputs.

The simplest use that can be made of a binary signal is in representing truth values (or any other 2-state domain). Logic gates provide the basic mechanism for manipulating data in binary form. Higher-order functions can be implemented by combining gates into circuits. Questions in this section will be strictly confined to the three logical operations AND, OR and NOT.

⁄ Evaluating a boolean expression (propositional logic formula) - such as NOT(A OR B) AND C given the values of the boolean variables, by converting to a logic circuit. Concept of a port as a set of logic AND gates in parallel.

⁄ Candidates should be able to convert a boolean expression into a logic circuit, and vice versa, and perform simple analysis of such a circuit using truth tables. Candidates are also expected to be able to design a logic circuit to implement a simple boolean function (presented either as a propositional expression or as a prose description).

NUMBER SYSTEMS

Representation of numbers in binary. The concept of a register. Operations of register, complementation, ranges, left and right shifts. Numerical overflow.

Conversion between decimal and binary, and between binary and hexadecimal.

The aim is to demonstrate how GROUPS of binary signals can be used to represent data which can assume more than just two values. For this reason, it is sufficient to demonstrate the principle using only non-negative integers. It is also important that the candidate understand the relation between the RANGE of values the data may assume, and the number of bits required to represent it. Hence, how numerical overflow may occur when adding or multiplying two numbers. Knowledge of octal is not required. Hexadecimal should be presented

ONLY as a shorthand notation for binary.

⁄ 2's complement representation. Binary addition and subtraction (by complementation and addition) in 2's compliment.	*⁄ For paper IIA, it is required that the candidate also knows how negative integers may be represented in binary. Although not directly mentioned in the syllabus, it makes sense to first discuss sign-magnitude representation before introducing 2's complement representation. Questions about addition and subtraction will however only be set using 2's complement.*
CODING SYSTEMS Range of possible symbols that can be represented by a given number of bits Representation of text using an 8-bit coding system (e.g. ASCII).	Candidates need to appreciate that groups of bits need not only represent numerical data, but can also represent textual data if an appropriate coding system is used. Some awareness of how the choice of coding system effects data portability is also required (e.g. problems which arise when transferring textual data between ASCII and EBCDIC machines).
⁄ Importance of collating sequence in a character-coding system (e.g. the ASCII code). Problems with representing international character sets.	*⁄ Much thought has gone into devising character codes which simplify text processing, for example sorting textual data in alphabetical order, distinguishing upper from lower case characters, etc. Candidates need to be aware of these considerations, as well as of the need for devising a character set which can handle all international languages as global data communication networks become a reality.*
2.3 THE CPU The processor and main memory. The main components of the processor - control unit, ALU. Registers (esp. accumulator) and program counter. Main memory and memory addresses. The address bus and how its width relates to the size of the address space. The data bus. The computer's word size (or word length). The instruction set as a means of controlling the CPU's circuitry. Function codes (opcodes). Operands. Concept of a machine code program as a set of instructions. Concept of stored-program. Concept of language levels (machine code being at a lower level than assembly language). Brief account of fetch and execute cycle.	The primary objective of this section is to give the candidate some idea of how the CPU functions - what instructions a processor typically understands, and how these instructions can be held in memory, together with the data to be operated on. No knowledge of assembly language is expected - simply an appreciation of how tedious programming in such a language can be. A demonstration of an assembly language program, together with the contents of the resulting executable, will help students understand the difference between assembly language and machine code.
Processor speed (cycles per second expressed in Hz, KHz, MHz, GHz). Units of time measurements: milli, micro, nano seconds.	Frequency units are Hz, KHz (kilo), MHz (mega), GHz (giga). Units of time are seconds, milliseconds, microsecond and nano seconds. It is important that students understand that frequency

units and units of time are directly related.

/ Typical machine code instructions - load, store and process (e.g. add, sub etc.) instructions. Using mnemonics to represent machine instructions. Symbolic addressing. Conditional and unconditional branches. Understanding, modifying and writing simple assembly-language programs.

/ Although candidates are not expected to have done any actual assembly-language programming (modern assembly languages are too complex), they should have some paper-and-pencil awareness of how very simple algorithms can be coded in an assembly language. Exposing candidates to some simple assembly language will help them to understand the concepts of language levels more than any amount of verbal description would. Moreover, the operation of a CPU cannot be meaningfully described unless candidates have some knowledge of an assembly language. Questions on this part will concentrate on the candidates' ability to UNDERSTAND and MODIFY a simple snippet of assembly language. Candidates will NOT be required to code algorithms in assembly language.

2.4 TRANSLATION OF HIGH-LEVEL LANGUAGES

Languages of a higher level than assembly language. The need for translators. Source code and executable code. Language translation as a transformation which preserves the semantics of a program - hence a high-level statement may result in many low-level instructions.

Candidates should appreciate how far removed high-level languages are from the machine code instructions understood by the CPU - and how richer high-level constructs are. Hence the one-to-many relation between a high-level construct and the equivalent low-level instructions. This can be demonstrated using natural languages - e.g. certain words in English may translate to whole phrases in Maltese. Samples of high-level source-code in different languages may be used to give students some idea of what programming in such languages is like. Candidates need not completely understand such samples in order to get the idea. It must be made clear that language translators are themselves programs. For the purpose of this syllabus, compilers directly produce executable code - NOT object code (since no mention will be made of program linkers and loaders). High-level language translation will be treated in more detail in Part III, within the context of a specific programming language.

2.5 THE OPERATING SYSTEM

System software as the layer between the user/application and the hardware. Resource management functions (e.g. managing the filing system).

Candidate should appreciate the role the Operating System, and system software in general, plays in a computer system. They should be familiar, through practical sessions, with hardware aspects of one operating system - file maintenance, loading and run programs, using system utilities, etc.

The O.S. as a communication medium between the units of the machine and the peripheral devices.

2.6 STORAGE

Main memory - volatile/writeable (RAM) and non-volatile/non-writeable (ROM), and uses of each (e.g. bootstrap loader in ROM). Secondary storage devices and media - magnetic media (floppy and hard disks, tapes) and optical media (CD-ROM). Uses of each, relative capacities and speeds, and access modes (direct vs. sequential). How a floppy disk-drive works - read/write heads, tracks, sectors.

There is no need to differentiate between ROM, PROM and EPROM. However, some mention of battery-backed memory for holding changeable configuration data is in order. The speed differences between main and secondary storage, and between floppy and hard disk drives, should be emphasised. Care should be taken when explaining the difference between random and sequential access – some candidates get the impression that only a tape drive is capable of sequential access. It is also important that candidates appreciate the SUITABILITY of storage devices for different requirements. The use of CD-ROMS for multimedia will be covered in more detail in Part 5.

/ The disk filing system - storage blocks, disk directory, file allocation. Hierarchical directory structure. Access time (typical,

/ Candidates should have an idea of how files may be stored on a disk – how a file is allocated a number of sectors, and how a directory keeps track of where each

faster or slower only).	*file is stored. For the purpose of this syllabus, it is sufficient to consider only contiguous file allocation. It is also important that candidates are aware of the problem of organising files on high-capacity media, and how hierarchical directory structures can simplify storage organisation.*

2.7 I/O DEVICES

INPUT DEVICES

Different types of input devices, and the application areas for which each is best suited. The QWERTY keyboard - alphanumeric and special keys. Pointing devices - the mouse, trackball, trackpoint, touchpad, light pen and touch screen. Graphics tablet, image scanner and digital camera. Bar-code reader, OMR and MICR. OCR, handwriting recognition and pen computing.

Candidates should be familiar with the characteristics and typical use of a number of input devices, and appreciate the advantages and disadvantages of each. Where possible, the use of each device discussed should be demonstrated. It is important that the candidate understand that although input devices are not under software control, the interpretation of the signals generated by such devices depends entirely on the software. The candidate is expected to be familiar with at least the following input devices and typical applications:

1. Keyboard (word processing)
2. Mouse (menu selection)
3. Joystick (games and simulations)
4. Bar-code Reader (point of sale)
5. Magnetic-Ink Reader (bank applications)
6. Scanner and OCR (converting hardcopy to machine-readable form; signature recognition)
7. Graphics Tablet (graphics application)
8. OMR (correction of multiple-choice examination questions)
9. Pen (handheld or very small computers on which it is physically impossible to fit a keyboard)

OUTPUT DEVICES

Distinction between hard-copy and soft-copy, and between vector (plotter) and raster (laser and dot-matrix printers and screen) devices. Resolution of a raster device and how this relates to both print quality and amount of data that needs to be transferred from the computer to the device (and hence speed).

Candidates need to be familiar with the operating characteristics and typical uses of the following output devices.

1. Daisy-wheel printer
2. Line printer
3. Dot-matrix printer
4. Inkjet printer
5. Laser printer
6. Plotter
7. VDU/Monitor

The difference in output quality can be easily demonstrated by providing students with samples - if applicable printed at different resolutions. Enlargement of a printout may also help to demonstrate the way the image is formed.

⁄ *How an image is displayed on a CRT, FPD (Flat Panel Display) or LCD.. Pixels and colour depth. Palettes.*	⁄ *Candidates should have some idea of the graphics subsystem of a computer. The material should be approached from an informative rather than a technical point of view. Little depth is expected.*
⁄ *A brief overview of the I/O Subsystem: the speed difference between the CPU/RAM and I/O devices. I/O buffering. Disk caching. Serial vs. parallel data transfer.*	⁄ *Like the graphics system, the I/O system should be approached from an informative point of view. Candidates need to be aware of how buffering reduces the speed difference between devices, and how caching can reduce the need to access slow devices frequently.*

2.8 DEDICATED COMPUTER SYSTEMS	
Difference between a general-purpose and a dedicated computer. Embedded and process control systems. Computerised appliances as an example of dedicated systems - VCR, auto pilot, mobile phones, GPS, etc. Specialised I/O devices (sensors, buttons, LCD). Software for dedicated computer systems.	*Candidates need to be aware that computers come in many forms, and that input and output peripherals need not be keyboards and screens but can be as simple as a button or a sensor, an LCD display or even a single LED. Embedded systems controlling common appliances are one form of dedicated computers - they have a CPU, I/O devices, software stored in ROM, and some RAM as workspace.*

PART 3
COMPUTER SYSTEMS

SYLLABUS	SUPPLEMENTARY NOTES
OBJECTIVES This section is meant to widen the candidates' view of computers and make them aware that some applications are too complex to be handled by a standalone desktop PC. It is recommended that candidates become familiar with the needs of systems such as airline reservation, police, hospital, finance and banking, weather forecasting and process control of industrial plants.	In section 2.5, candidates familiarised themselves with the use of one specific operating system. In this section, they should be made aware of the existence of other types of operating systems and their characteristics - single-user vs. multi-user, networked systems, single-programming vs. multi-programming. It is strongly recommended that candidates be shown around a large computer installation. It is important that students understand the need for the different types of operating systems discussed - in other words, that different types of operating systems seek to fulfil different requirements. No knowledge of the internal workings of an O.S. is expected.
3.1 PROGRAMMING AND APPLICATION PACKAGES Off-the-shelf, customisable, and tailor-made packages. Advantages and disadvantages of each. Software licensing considerations.	Students also need to be aware of different user licensing e.g. freeware, shareware, site licensing, upgrades, patches, software registration etc. The process of installing software and configuration as well as compatibility considerations, technical support and anti-piracy

features.

/ *Awareness of the importance of choice of programming language in developing application software. Awareness of 4th generation languages - demonstration of designing a database.*

3.2 ROLES RELATED TO AN I.T. ENVIRONMENT

Knowledge of the existence of a wider range of tasks and responsibilities and hence the need to share them. The ability to outline the duties of:

1. Information Systems Manager
2. Systems Analyst/Designer
3. Systems Administrator
4. Programmer
5. I.T. Trainer
6. Operator
7. Data Clerk
8. Web Master
9. Lab technician
10. Maintenance engineer

Candidates need to appreciate the fact that there is no hard-and-fast rule about how a DP department is organised, and that this depends very much on the size and complexity of the system to be managed. In smaller departments, some of the roles may be the responsibility of a single individual.

3.3 SYSTEMS ANALYSIS

Awareness of the process of analysing a system with a view to computerisation:

1. Project selection and feasibility study,
2. Present system study and analysis,
3. Design of new computerised system,
4. Programming and documentation,
5. Implementation and changeover methods,
6. Control and review
7. System maintenance

Systems analysis is so far removed from candidates' everyday experience that unless it is introduced using an example system it will make little sense to them. A system should be carefully selected in order to ensure that all (or at least most of) the steps involved in systems analysis can be meaningfully demonstrated. A lending library system is ideal for this exercise, because candidates are familiar with the manual operation of a library.

A stock control example can be used to demonstrate system analysis.

3.4 NETWORKS

Networking: LAN and WAN. The advantages of a LAN over a number of standalone PC's - sharing of hardware and software resources, ease of management/control by the system administrator.

The use of MODEMS to interface computers with telecommunications networks (telephone cable, optic fibre, microwave, satellite links).

Computer communications over WANS; e-mail, WWW, Video Conferencing.

It is important that candidates understand the concept of a resource, and of shareable and non-shareable resources. Ideally, candidates should have had experience with using a LAN (most school labs are equipped with a networked system). This makes it easier for candidates to appreciate many concepts which are otherwise hard to grasp if one's experience is limited to using a standalone computer. The relative advantages and disadvantages of networked and standalone systems should be highlighted. Mention should also be made of distributed databases as an example of sharing non-centralised data resources.

LANs can be connected to form WANs, possibly spanning international borders, using diverse communications means - telephone lines, microwave, satellite etc. Candidates should also understand the importance of a modem to convert from digital to audio signals for transmission over

	voice networks, and vice-versa.
⁄ Server and client machines; the problem of bandwidth at a general level.	*⁄ General idea that the size of data to be transferred, the time taken for the transfer and bandwidth are directly related*

3.5 TYPES OF OPERATING SYSTEMS

⁄ Real-time, batch, time-sharing (on-line) use. Their meaning - differences shown by examples. The suitability of each operating mode for a particular application.	*⁄ Batch processing should not be introduced as a historical operating mode, but as a mode which still has its uses and may be the preferred mode of operation under certain circumstances. Batch processing may be easily demonstrated using the operating system shell's script language.*
⁄ The necessity of different OS's to support modes and configurations mentioned in 3.4. Common types of operating systems - single-user, multi-user, networked, single programming, multi programming.	*Awareness of the fact that in large computer systems the OS must be able to manage:* *1. a wider range of hardware resources* *2. more users* *3. a greater number of utilities* *It is sufficient for candidates to be able to give examples of these.*

PART 4
ALGORITHMIC PROBLEM SOLVING AND PROGRAMMING

SYLLABUS	SUPPLEMENTARY NOTES
OBJECTIVES The candidate should:	
1. understand the task a program is required to perform 2. design an appropriate algorithm to perform the task 3. code the algorithm in a high level language 4. demonstrate an awareness of structured programming techniques 5. test the program for its performance according to the specifications	Candidates must have access to a compiler or interpreter for the language of choice, as well as to utilities appropriate to the job of editing, compiling, running and debugging a program (text editor, linker, etc.). This section should encourage teachers to create a constructive learning environment in which students develop concepts, planning and procedural thinking skills.
A working knowledge of either BASIC or PASCAL is required. Candidates are free to use either of these languages in answering questions. Questions testing candidates' reading knowledge of a language will be set in both BASIC and PASCAL. Candidates should have practical experience in the form of programming exercises which involve the use of a computer. They should be encouraged to use, criticise and modify a number of programs written by other people.	If the language of choice is BASIC, it is recommended that a structured version such as QUICK BASIC is used. If the language of choice is PASCAL, a UCSD-type Pascal (with support for string variables) should be used.

4.1 DEFINITION AND ANALYSIS OF PROBLEMS

Students should be able to analyse the requirements of a problem and create specifications and target for the solution.

Specifications should concentrate on inputs and outputs.

o DESIGNING A SOLUTION TO THE PROBLEM

The development of algorithms.

The use of flow charts for describing an algorithm.

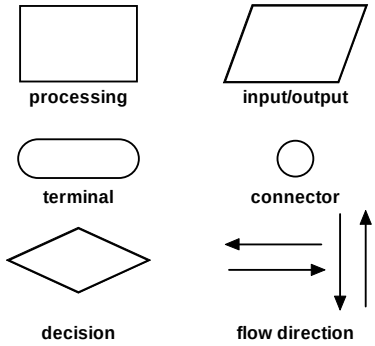
The use of structured programming utilising a top down approach should be emphasised from the beginning.

Pseudo-coding and flowcharting are suitable methods for designing the solution of a problem as we apply the top down method.

Flowchart symbols

Teachers are advised to restrict themselves to a limited number of simple conventional symbols for both program flowcharts and system flowcharts. Program flowcharts should be language independent.

Questions set will be restricted to the following six symbols:



/ The use of structure diagrams and pseudo-code for describing an algorithm.

/ Pseudo-code should be used in cases where flowcharts are too cumbersome. The following keywords for developing pseudo-code are suggested:

***Arithmetic**: add, subtract, multiply, divide, calculate*

***Data transfer**: move, store, replace*

***Input/output**: read, input, output, print*

***Decision**: if...then...else*

***Repetition**:repeat...until,while…endwhile, for…end for.*

- **FEATURES OF PROGRAMMING LANGUAGES**

  Data types. Constants and variables, expressions and assignments, operators and operator precedence, input and output statements, built-in functions, sequential and conditional execution, looping constructs. Single dimensional arrays.

Initially, students should not be required to write programs from scratch, but should be provided with ready-written source code which they are to modify and/or debug. This helps the novice programmer get to grips with both the editor and the language.

The following constructs should be covered:

1. Printing and formatting of output:

   Basic: Print, Tab
   Pascal: Write, Writeln, colon formatting of both integers and reals

2. Assignment statement, arithmetic expressions and operators:

   Basic: *Let, +, -, *, /, ^*
   Pascal: *:=, +,* **-, *, /, Mod, Div**

3. Inputting data from the keyboard:

   Basic: *Input*
   Pascal: *Readln*

4. Loops (including nested loops):

   Basic: *For..Next, Do..Loop Until, Do While…Loop*
   Pascal: *For..To/Downto..Do, Repeat..Until, While..Do*

5. Conditional transfer:

   Basic: *If..Then(..Else)*
   Pascal: *If..Then(..Else), Case..Of*

6. Declaring arrays (both numeric and string). Reading data into an array.

7. Declaring and calling functions and procedures/subroutines. Parameters (input only).

   Standard functions and procedures:

   Basic: Abs(x), Fix(x), Sqr(x), Int(x)
   Pascal: Abs(x), Sqr(x), Sqrt(x), Int

8. Random numbers - applications.

   Basic: Randomize (Timer), RND ()
   Pascal: Randomize, Random()

9. String variables and string handling. Comparing strings. String concatenation.

   String functions (such as length, substring extraction, conversion between strings and numbers).

   Conversion between character and ASCII.

   Basic: *Left$(x$), Right$(x$), Mid$(x$), Chr$(X$), Asc(x$), Val(x$), Len(x$), Str$(x$), Inkey$*
   Pascal: *Copy(str,from,to),Length(str), Ord(chr), Chr(int), Val(str,num), Str(num,str)*

   Note: Pascal syntax might vary depending on compiler.

⁄ *Procedures, subroutines and user-defined functions as available in the language of choice. Local and global scope. Two-dimensional arrays.*	⁄ *Candidates are expected to be able to write simple procedures and functions with parameters. Only value parameters need be considered. It is very important that candidates understand the concept of a scope. File handling should be limited to text files. Candidates are expected to be familiar with commands/standard procedures to create and open files, read and write text files sequentially,close files. The commands/ procedures required to handle files will of course vary depending on the language of choice.*
⁄ *Simple text file-handling. Reading and writing to a text file.*	

4.4 TESTING

Types of errors: syntax, logical, run-time. Candidates should be able to distinguish between them and give examples. Appropriate methods of detection and correction should be understood.	Candidates should be able to identify and correct various types of errors using appropriate diagnostic aids.
Use of suitable test data to check the performance of a program. Output of intermediate results. Dry running. Program tracing.	Candidates should devise and use suitable test data to check that a program performs according to specification, including input of incorrect data by the user.

4.5 DOCUMENTATION

The difference between user, technical and program documentation. Candidates should be able to describe and give examples of essential features found in user documentation accompanying software packages.	Candidates will be required to state the items likely to be present in each of the types of documentation and why each of these items are required. Candidates should have access to user manuals of some common application packages.

4.6 LANGUAGE TRANSLATORS

The difference between compilers, interpreters and assemblers - the relative advantages (and disadvantages) of each. Source code and executable code. Error messages.	Candidates should be aware of the use of different sets of machine code instructions on different machines and of the common types of operation: arithmetic, logical, fetch and store, branch and input/output. Candidates should be able to understand the simple concepts involved in the functioning of hardware at machine code level. With regards to assembly language, candidates should appreciate the importance of symbolic addressing for both instructions and data, and the value of mnemonics. The idea of a one-to-one correspondence between an assembly language and a machine code instruction should be understood. It is not expected that candidates have practical experience in programming in an assembly language. The facilities offered by the various types of translators should be compared. This is best done by means of a demonstration illustrating the basic features such as ease of use, object code, and error messages.

⁄ *4GL's. Software portability.*	⁄ *Candidates should have some idea of how 4GL languages differ from 3GL languages, and their advantages (higher-level, automatic code generation, etc.) and disadvantages (size and speed penalty). Although no practical experience of using a 4GL is expected, students should be given a demonstration of some application written using both a 3GL and a 4GL so that they can clearly see the difference. Candidates also need to be aware of the desirability of software portability, and how this relates to language levels.*

PART 5

INFORMATION AND COMMUNICATIONS TECHNOLOGY IN SOCIETY

SYLLABUS	SUPPLEMENTARY NOTES
OBJECTIVES	
The candidate should: • be knowledgeable of the major areas of computer applications, • be aware of the vast amount and variety of computer software available for different applications. • have an understanding of the practical problems in using computers	Candidates should have access to suitable computer facilities supporting a number of application packages.
o AREAS OF COMPUTER APPLICATIONS	
The range of applications familiar to the candidate should include:	The aim is to give the candidate a knowledge and understanding of a wide range of applications. The general principles are more important that a detailed knowledge of particular equipment. Excursions to places where students can appreciate the use of computers in different spheres of work and leisure would be highly motivational. Candidates should be familiar with the general concept of forms of applications in particular the following should be emphasised: global networks as the Internet, and the possibility of browsing this set of networks, using it as a source of information.
• commercial data processing: e.g. stock control, reservations, administration, POS systems.	
• technical, mathematical and scientific uses: e.g. medical diagnosis, CAD, simulation, weather forecasting.	
• Communication and information systems: Internet and WWW, electronic mail.	
• in industry: computer process control: e.g. industrial processes, robotics, gas and oil exploration, monitoring and using energy, CAD-CAM	CAD-CAM (Computer Aided Design – Computer Aided Manufacturing)
• educational uses: e.g. CAL, interschool projects using Internet, school administration	CAL (Computer Aided Learning); CBT (Computer Based Testing)
• leisure and home uses: e.g. games, microprocessor-controlled home appliances, central heating	
• office automation: word processing, database systems, spreadsheets, creation and use of graphics presentation software, personal organizers and schedulers.	
• finances: shops, banks, EFT, stock control, supermarkets, stock exchange, insurance, e-commerce	EFT (Electronic Fund Transfer)

- travel: air traffic control, train control, ocean navigation, 'intelligent' cars, space travel.
- In the community: police, health, schools, ecological interests, libraries, supermarkets, teleshopping.

5.2 EFFECTS OF COMPUTER-BASED SYSTEMS ON INDIVIDUALS, ORGANISATIONS AND SOCIETY

Positive and negative effects of computerisation, e.g. satisfaction and efficiency at work, effects of computer games on youngsters, health hazards from working long hours at a computer, opportunity for crime.

Computers are machines which as much as they can be useful, can also have negative effects.

5.3 DATA SECURITY AND PRIVACY

Need of a data security and integrity of data. Backups (the generations of files: grandfather, father, son files), physical security and software safeguards.

Candidates should be aware of methods used for ensuring privacy and integrity of data. The size and usage of computer systems make different backup demands. For example a mainframe computer system of a bank requires different and more sophisticated backup provisions than a small microcomputer system used by a shop owner for stock control purposes.

Software piracy and copyright. Ethical and legal issues. Hardware and software procedures which deter piracy – serial numbers and activation keys, hardware keys (dongles). Software registration.

Physical security includes both physical facilities to prevent accidental loss such as use of write protect tabs and facilities to deter purposeful corruption and fraudulent use of data, such as restricted access to computer areas. Software facilities include use of IDs and passwords, data encryption, log and audit trail.

Unauthorised copying of software is detrimental for software houses whose existence depends on the sales of their products. This international problem has given rise to laws and hardware and software protection aimed at curbing the crime. Students should be aware of the above problem, security measures and their consequences.

(*Access rights. Privacy on multi-user/network systems.*

o MULTIMEDIA

Brief overview of capabilities and trends. Future perspectives (home office, access to public and institutional databases, libraries, high-quality sound and pictorial data representation). Basic hardware and software requirements and costs.

Candidates are required to have an awareness of this technological development. Though more still in experimental stages applications of multimedia are limited only by human imagination. Especially useful for international conferencing and to aid handicapped people.

APPENDIX 1

COURSEWORK

These are the coursework exercises prescribed for the SEC in Computer Studies. Candidates are required to present all four exercises mentioned below. All coursework MUST be word processed.

Section 1 ALL candidates must present this exercise.

Section 2 Candidates opting for paper IIB are to present exercise 1 from this section, while those opting for paper IIA are to present exercise 2.

Section 3 ALL candidates must present this exercise.

Section 4 ALL candidates must present this exercise.

MARKING

The coursework carries a maximum of 150 marks, which account for 15% of the assessment. The raw mark obtained by a candidate in the coursework will be scaled down by a factor of 10.

SECTION 1 – SPREADSHEETS

The student is to produce a spreadsheet of about one/two screens of a normal spreadsheet package (such as Lotus 123, Multiplan, Excel or Quattro Pro) on a topic such as the ones suggested below or one approved by the teacher who should make sure that it is within the reach of the student and understandable to him or her:

- weekly sales of several shops (to show totals by week, by shop, highest/lowest/average sales, commissions paid, profit etc.)
- a simple balance sheet
- an invoice (to include discount, VAT, deposit, sub-totals)
- hire purchase (over a specific period of say 3 years – to show monthly instalments, interests, etc.)
- a family budgeting spreadsheet per month with a projection for a one year period – to show incomes, various expenses, balances;
- a physical or mathematical model involving data graphing;
- any other spreadsheet as approved by the teacher as long as it is within the knowledge of the student and of a level as the ones indicated above
- student class marks for an examination including total, maximum, minimum and average.

Criteria for Assessment:

Marks are to be awarded for a report prepared by the student according to the following scheme:

a. Definition of the problem tackled – advantages and disadvantages of using a spreadsheet in comparison to using any other application to solve the chosen problem. [4]

b. Implementation of spreadsheet tools used to solve the problem describing the use of:

- the use of formulas; [1]
- the use of simple functions such as =SUM, =MAX, =MIN or their equivalents; [1]
- the ability to copy cells; [1]
- the ability to move cells; [1]
- insertion of rows and columns; [1]
- naming a range of cells and using it in formulas; [1]

- adjusting column widths; [1]
- printing a range of cells; [2]
- a simple one column sort. [2]
- a simple replicate command; [2]

The ability to make use of the above features must be either demonstrated or explained.

c. Printouts of spreadsheets showing test date supplied by the teacher. [4]

Printout of spreadsheet showing formulae. [4]

d. Comments and conclusions [2]

Total Marks: [25]

SECTION 2 – PROGRAMMING EXERCISES

This exercise tests the candidate's competence in:

1. Specifying the task which the program is required to perform
2. Formulating a structured algorithm and describe it by means of a suitable diagrammatic methodology pseudo-code.
3. Implementing the algorithm in a high-level programming language using structured techniques.
4. Testing the program using suitable test data.
5. Writing documentation, including user instructions and description of simple data types and structures used.

Notes:

- Candidates are to present only ONE of the following programming exercises, depending on choice of Paper IIA or Paper IIB.
- Any high-level language specified in Part 4 may be used, as long as it enables the candidate to meet the above assessment requirements.
- It is expected that teachers give guidance and suggestions to all candidates in the choice of programming tasks, relating to their ability as well as to hardware and time constraints. Students are free to come up with their own ideas regarding programming tasks, subject to approval by their teacher.

Exercise 1: (This exercise is to be attempted by candidates opting for paper IIB).

A program which makes use of all the following constructs;

- input and output statements
- assignment
- conditional execution and branching
- loops
- string handling

Exercise 2: (This exercise is to be attempted by candidates opting for paper IIA)

In addition to the above, the program must exhibit a modular structure (user-written subprograms or procedures/sub-routines) and make use of single-dimensional arrays.

Documentation:

1. Define the task the program is to perform, including input and output requirements.
2. Provide instructions for a user to run and operate the program.
3. Provide a description of the algorithm and a hard-copy listing the program.
4. Provide a description of data types and structures used by the program.
5. Provide evidence, in the form of test-run printouts and screen dumps, that the program has been fully tested.
6. Comments and conclusions.

Criteria for Assessment:

1. Definition of the problem [12]
   A statement indicating the scope of the problem to be tackled.
   A statement of the results required.
   Details of the input information required.
2. The solution of the problem: [12]
   a flowchart of the algorithm
   computer listings of original or modified program
   details of any special design features
3. Running the program: [8]
   evidence that the solution works, e.g. screen dumps
   details of test data
4. User instructions: [4]
   loading the program
   running the program
5. Comments and conclusions: [4]
   limitations of the program
   possible improvements
   evaluation of success or otherwise of program as implemented

Total Marks: [40]

Examples of Programming Tasks Which May Be Set

7. An examination has been administered to a class of students and the scores for each have been provided as data along with the students' names. Write a program to do the following:

   - Assign a grade to each student based for example 4 scale boundaries.
   - Determine and print the class average.
   - Determine the first in class; the highest and lowest mark.

   - Calculate the number of cases obtaining each grade and draw a histogram using the asterisks.
   - Print a list of the results.

8. Write a program which plays the game of HANGMAN. Read each letter of the word to be guessed into successive elements of the string array W$. The player must guess the letters in W$. The program should terminate when either all letters have been guessed correctly (player wins) or a specified number of incorrect guesses have been made (computer wins).

   HINT: Use a string array S$ to keep track of the solution so far. Initialise each element of S$ to the character "*". Each time a letter in W$ is guessed, replace the corresponding "*" in the array S$ with that letter.

9. Write a program which assists the learning of a topic in a school subject – Maths, Physics, History, Italian, etc.

10. Write a simple stock-control program having the following features. The stock records are to be held in an array rather than in a file.

a. produces and invoice;

b. updates the quantity in stock;

c. searches for a particular item;

d. produces an inventory list;

e. produces a suppliers' list.

11. A simple payroll with inputting of e.g. hours worked, overtime hrs, rates for Income Tax and N.I., allowances, deductions etc, and printing of simple payslips including the gross and net pay.

SECTIONS 3 – SYSTEM ANALYSIS

The candidate is expected to investigate the possibility of computerising a manual system accessible to the candidate. Examples of suitable systems are:

- stock control (e.g. school store)
- school lending library
- video shop
- examination processing system
- school tuck shop
- CAL tool for a particular area of the curriculum
- any other system subject to approval by the teacher

Criteria for Assessment:

The candidate is required to:

1. Give a clear problem definition [5]
2. Give background investigation [15]
   - prepare a set of questions to ask user
   - describe manual system
   - list drawbacks of manual system
   - explain why a commuter solution is appropriate
   - specify system requirements

3. Propose new system

   hardware specification [4]

software specifications: [12]

- identification of possible application packages (if applicable)
- input and output requirements
- operations required (grouped functionally into menus and submenus)
- list of necessary files and record structures. Candidates are not expected to define types and fields required

4. Specify methods of testing [3]
5. Describe change-over procedure [1]

Total Marks: [40]

SECTION 4 – DATABASES

Candidates are expected to develop a simple database using the application package with which they are familiar. The following are some examples of databases which candidates at this level should be able to attempt:

- Bus Routes
- Food
  - recipes
  - food nutrition
- Famous people:
  - actors
  - singers
  - authors
  - inventors
  - explorers
- Music
  - composers
  - pop hits
  - classical music
- Simple stock control
- Medical records
  - hospital
  - clinic
- Simple payroll
- Animal (classification of)
- Countries
- Elements (chemical
- Simple book/video library
- Student inventory
- Student particulars
- Historical sites in Malta
- Sports personalities
- Astronomy
- Bank control
- Collections: any database on collections of stamps, sea shells etc.

Candidates are required to produce a report:

- describing the structure of the database (files, field specifications giving field names, types and lengths);
- explaining the use of update facilities;
- providing hardcopy views of different formats and queries performed.

Criteria for Assessment:

1. File Structure:
   - File specifications (field name, field type, field size) [2]
   - A minimum of 20 records [1]
   - Record to include a minimum of 5 fields [1]
   - Field types to include character, numeric and date [1]
2. Update Facilities:
   - appending [2]
   - editing [2]
   - deleting records [2]
3. Query Facilities
   - listing /printing of data in a least TWO different formats [2]
   - simple query including one condition [2]
   - query involving a compound condition [2]
4. Database Manipulation
   - Generate a report including a simple calculation on a field or group of fields (this may involve exporting data to a spreadsheet and generate the required report using spreadsheet facilities) [4]
   - Sorting the database by at least two criteria [4]

Total Marks: [25]

SECONDARY EDUCATION CERTIFICATE IN COMPUTER STUDIES

TEACHER-ASSESSED PRACTICAL SCORE

Coursework Marking Scheme

Section 1

SPREADSHEETS

Criteria for Assessment	**Maximum Mark**	**Actual Mark**
definition of the problem tackled, comparison with using any other application to solve the chosen problem.	4	
Implementation of spreadsheet tools used to solve the problem describing:		
the use of formulae	1	
the use of simple functions	1	
the ability to copy cells	1	
the ability to move cells	1	
insertion of rows and columns	1	
naming a range of cells and using it in formulas	1	
adjusting column widths	1	
printing a range of cells	2	
a simple one column sort	2	
a simple replicate command	2	
Proof of completion of exercise and testing:		
Printouts of spreadsheet showing test date supplied by the teacher.	3	
Printouts of spreadsheet showing formulae	3	
Comments and conclusions	2	
TOTAL	**25**	

SECONDARY EDUCATION CERTIFICATE IN COMPUTER STUDIES

TEACHER –ASSESSED PRACTICAL SCORE

Coursework Marking Scheme

Section 2

PROGRAMMING

Criteria for Assessment	**Actual Mark**	**Maximum Mark**
Definition of the problem:		
statement indicating the scope of the problem to be tackled	4	
statement of the results required	4	
details of the input information required	4	
Solution of the problem:		
algorithm (flowchart or pseudocode)	4	`
computer listing of the original or modified program	4	
details of any special design features	4	
Running the program		
evidence that the solution works (e.g. screen dumps)	4	
plan and evidence of test data	4	
User instructions:		
How to load the program	2	
How to run the program	2	
Comments and conclusions	4	
TOTAL	**40**	

SECONDARY EDUCATION CERTIFICATE IN COMPUTER STUDIES

TEACHER-ASSESSED PRACTICAL SCORE

Coursework Marking Scheme

Section 3

SYSTEM ANALYSIS

Criteria for Assessment	Maximum Mark	Actual Mark Mark
Problem definition	5	
Background investigation:		
prepare a set of questions to ask user	3	
describe manual system	3	
list drawbacks of manual system	3	
explain why a computer solution is appropriate	3	
specify system requirements	3	
Proposal of the new system:		
Hardware specifications	4	
Software specifications		
• identification of possible alternative application packages	3	
• input and output requirements	3	
• operations required	3	
• list of necessary files and record structures	3	
Specification of methods of testing	3	
Description of changeover procedure	1	
TOTAL	**40**	

SECONDARY EDUCATION CERTIFICATE IN COMPUTER STUDIES

TEACHER-ASSESSED PRACTICAL CODE

Coursework Marking Scheme

Section 4

DATABASES

	Maximum	Actual

Criteria for Assessment	**Mark**	**Mark**
File structure		
file specifications	2	
Minimum requirements:		
file of 20 records	1	
records to have a structure of 5 data fields	1	
field types to include character, numeric and date	1	
Use of update facilities:		
appending a record	2	
editing a record	2	
deleting a record	2	
Use of query facilities:		
printing of data in 2 different formats: table view form view	1 1	
simple query including one condition	2	
query involving compound condition	2	
Manipulation of database:		
generation of report	4	
sorting database	4	
TOTAL	**25**	

SECONDARY EDUCATION CERTIFICATE IN COMPUTER STUDIES

TEACHER-ASSESSED PRACTICAL SCORE

Coursework Marking Scheme

WORDPROCESSING

Assessed with reference to the overall presentation of the other 4 sections of the coursework.

Criteria for Assessment	Maximum Mark	Actual Mark
Good page layout	2	
Underlining	2	
Text Centering	2	
Indenting paragraphs	2	
Right margin justification	2	
Use of tables	2	
Use of bold, italics, and/or different fonts in different sizes	2	
Inclusion of headers and footers	2	
Page numbering	2	
Use of bulleted and/or numbered lists	2	
TOTAL	**20**	