Areas of Computer Applications

The range of applications should include:

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, e-government.

In industry: computer process control: e.g. industrial processes, robotics, gas and oil exploration, monitoring and using energy, CAD-CAM Educational uses: e.g. CAL, inter-school projects using Internet, school administration.

Leisure and home uses: e.g. games, microprocessor-controlled home appliances, air-conditioning, security systems.

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, ecommerce.

Travel: air traffic control, navigation (e.g. GPS), ‘intelligent’ cars, space travel.

In the community: police, health, schools, ecological interests, libraries, supermarkets, teleshopping.

Systems Analysis and SDLC
The process of analyzing a system with the view to computerization:

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

- Roles in an IT 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:

Systems Analyst/Designer
Programmer
I.T. Trainer
Data Entry Clerk
Web Master
Computer technician
Computer engineer

- Networks

Networking: LAN, MAN, WAN and WLAN as a variation of a LAN.

Advantages of a LAN over a number of standalone PCs – 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; email, WWW, Video Conferencing.

Server and client machines.

The problem of bandwidth at a general level.

- Project:
Operating systems

*Interfaces - CLI and GUI.*

System software as a layer between the user/application and the hardware.

Resource management functions; management of files, memory, CPU, I/O.

Real time, batch, time sharing (on-line) use.

Their meaning – differences shown by examples.

The suitability of each operating mode for a particular application.

Common types of operating systems – single-user, multi-user, networked, single programming, multi programming.

Difference between a general-purpose and a dedicated computer.

Embedded and process control systems.

Computerised appliances as examples of dedicated systems – VCR, auto pilot, optical recorder/player, mobile phones, GPS, etc.

Specialised I/O devices (sensors, buttons, LCD).

Software for dedicated computer systems.
• Low Level Languages, Language Translators

Languages of a higher level than assembly language. The need for translators.
Language translation as a transformation which preserves the semantics of a program – hence the high-level statement may result in many low-level instructions.
The difference between compilers, interpreters and assemblers.
The relative advantages and disadvantages of each.
Source code and executable code.
Error messages.
4GLs.

Software portability
Awareness of the importance of choice of programming language in developing application software.
Awareness of 4th generation languages – demonstration of designing a database.
Software licensing considerations.
Concept of language levels (machine code being at a lower level than assembly language).
The instruction set as a means of controlling the CPU’s circuitry.
Function codes (opcodes) and operands.

Concept of a machine code program as a set of instructions.

<table>
<thead>
<tr>
<th>Assembly Language</th>
</tr>
</thead>
</table>

Typical machine code instructions - load, store and process instructions.

Using mnemonics to represent machine instructions.

Immediate, direct and symbolic addressing.

Conditional and unconditional branches.

Understanding and modifying simple assembly language programs

| MATSEC Past Papers |