



L-Università  
ta' Malta

**MATSEC**  
Examinations Board



# IM 07 Syllabus

## COMPUTING

**2028**

## Table of Contents

Introduction .....	3
List of Subject Foci.....	3
List of Learning Outcomes.....	4
Scheme of Assessment.....	32
Appendices.....	35

## Introduction

Computing is a field of academic study that has engaged scholars and professionals worldwide for decades, and its innovations have become integral to modern society, driving advancements across numerous sectors.

The main purpose of this syllabus is to provide a solid understanding of the core principles and foundational concepts of computing, supported by essential technical knowledge that will help students gain meaningful experience in the field. This will adequately prepare them for future studies and careers in computing and related disciplines. The syllabus also offers learners the opportunity to explore the vast potential and challenges that computing presents to society, allowing them to critically assess both the opportunities and ethical considerations that come with technological advancements.

The syllabus assumes prior knowledge of computing at the level equivalent to the Secondary Education Certificate, building on this foundation to advance students' understanding. It is organized into several Learning Outcomes (1 – 8), each with corresponding Assessment Criteria to guide teachers, students, and examiners in determining the depth and breadth of the content to be covered.

While teachers may extend lessons beyond these outcomes, assessments will be based on the content outlined in the syllabus. The course aims to provide both theoretical knowledge in computing and practical skills, ensuring students are well-prepared for further academic study in the field. Regarding practical computing, students will learn to apply their skills responsibly, taking into account ethical considerations, data security, and the environmental impact of technology. Teachers should guide students in navigating these issues, ensuring their safety and promoting a responsible approach to technology use.

## List of Subject Foci

1. Digital Logic
2. Computer Architecture and Assembly Language
3. Operating Systems
4. Networking
5. Programming in Python
6. Systems Analysis and Design
7. Databases
8. Algorithms and Artificial Intelligence

## List of Learning Outcomes

At the end of the programme, I can:

- LO 1. relate the binary number system to digital arithmetic and logic operations to produce meaningful results.
- LO 2. explain how different hardware components interoperate in a computer system and how these can be programmed using Assembly.
- LO 3. explain the functions of different operating systems, how they manage memory, handle file systems, and process input and output operations.
- LO 4. develop a comprehensive understanding of networking concepts, technologies, and their practical applications in various contexts.
- LO 5. design, write, test, and debug Python programs using structured and object-oriented programming techniques to solve real-world problems effectively and efficiently.
- LO 6. analyse and design systems using established methodologies.
- LO 7. understand foundational database concepts, including structures, DBMS role, three-level architecture, relational model, E-R modelling, normalisation, and basic SQL.
- LO 8. represent algorithms using flowcharts and pseudocode, implement searching and sorting algorithms, and explain the technology behind artificial intelligence.

**Subject Focus: Digital Logic****Learning Outcome 1: I can relate the binary number system to digital arithmetic and logic operations to produce meaningful results.**

Topic	Sub-Topic	Assessment Criteria
1.1 Number Systems	1.1.1 Number bases	1) Define number systems. <i>Limited to binary, decimal and hexadecimal.</i> 2) Identify the purpose of using number systems to represent data and instructions. 3) Convert between binary, decimal and hexadecimal. 4) Represent the number of bits in a given register to $2^n$ where n is the number of bits in a given register.
	1.1.2 Units of information	5) Define a bit, nibble, byte, kilobyte (KB), megabyte (MB), gigabyte (GB) and terabyte (TB). 6) Convert between the units of storage mentioned in (5). 7) Differentiate between kibi (Ki) and kilo, mebi (Mi) and mega, and gibi and giga.
1.2 Binary Number System	1.2.1 Binary Numbers	8) Identify the purpose of using binary to represent data and instructions. 9) Define unsigned binary numbers. 10) Define signed binary numbers using two's complement representation. 11) Differentiate between unsigned and signed binary numbers.
	1.2.2 Binary Operations	12) Carry out binary addition. 13) Represent a number using signed register two's complement representation. 14) Carry out binary subtraction.
	1.2.3 Range of Numbers	15) Calculate the minimum and maximum values that can be represented in a given unsigned binary number (0 to $2^n-1$ ) where n is limited to 16.

Topic	Sub-Topic	Assessment Criteria
		16) Calculate the minimum and maximum values that can be represented in a given signed binary number using two's complement representation ( $-2^{n-1}$ to $2^{n-1} - 1$ ) where n is limited to 16.
	1.2.4 Binary Coded Decimal (BCD)	17) Define BCD. 18) Carry out conversions between BCD and Binary numbers and vice versa.
1.3 Fixed-point	1.3.1 Fixed-point binary numbers	19) Represent signed fractional fixed point binary numbers using two's complement. 20) Identify the range of signed fractional fixed point binary numbers using two's complement. 21) Carry out conversions to and from decimal numbers. <i>Limited to 8 bits for the integer part and 4 bits for the fractional part.</i>
1.4 Errors	1.4.1 Errors in Computer Arithmetic	22) Define the arithmetic errors introduced due to register (space) size. <i>Limited to overflow, underflow, and errors introduced when converting from decimal to binary.</i>
1.5 Character sets	1.5.1 Character Set Representation	23) Define the term 'character set'. 24) Interpret the relationship between the number of bits per character in a character set, and the number of characters which can be represented, e.g.: ASCII and Unicode (such as UTF-8). <i>Limited to 8 bits when using ASCII and 16 bits when using Unicode.</i> 25) Outline the difference between ASCII compared to UTF-8 as used in current systems.
1.6 Logic Gates	1.6.1 Basic Functions	26) Identify the functions of NOT, AND, OR, NAND, NOR, XOR and XNOR gates, including the binary output produced from all the possible binary inputs (all gates, except the NOT gate, will have 2 or more inputs). 27) Define a logic gate and/or a logic circuit and/or a truth table.

Topic	Sub-Topic	Assessment Criteria
		<p>28) Represent the AND, OR, NOT, NAND, NOR, XOR, and XNOR gates using standard symbols.</p> <p>29) Draw the truth table for a given logic circuit and/or Boolean expression. <i>Limited to 4 inputs using the AND, OR, NOT, NAND, NOR, XOR, and XNOR gates.</i></p> <p>30) Derive the truth table to represent the solution for a given scenario.</p> <p>31) Represent the AND, OR, NOT, NAND, NOR, XOR, and XNOR gates using Boolean expressions.</p> <p>32) Draw the logic circuit for a given truth table and/or Boolean expression. <i>Limited to using AND, OR, NOT, NAND, NOR, XOR, and XNOR gates.</i></p> <p>33) Derive the logic circuit to represent the solution for a given scenario. <i>Limited to using AND, OR, NOT, NAND, NOR, XOR, and XNOR gates.</i></p> <p>34) Express the Boolean expression from a given logic circuit and/or truth table. <i>Limited to those listed in (26).</i></p> <p>35) Derive the Boolean expression to represent the solution for a given scenario. <i>Limited to those listed in (26).</i></p> <p>36) Complete a given logic circuit and/or truth table and/or Boolean expression. <i>Limited to those listed in (26).</i></p> <p>37) Interpret a given logic circuit and/or truth table and/or Boolean expression. <i>Limited to those listed in (26).</i></p>
1.7 Karnaugh Maps (KM)	<p>1.7.1 Simplification of Boolean expressions</p> <p>1.7.2 Application of KM</p>	<p>38) Simplify two or three variable KM. <i>Candidates can be asked to simplify a KM using either sum of products (SOP) or products of sum (POS) (1's or 0's respectively) limited to up to 3 variable KM.</i></p> <p>39) Interpret don't care conditions</p>

Topic	Sub-Topic	Assessment Criteria
		40) Use don't care conditions to simplify KM.
	1.7.3 Half Adder and Full Adder	41) Define half adder. 42) Produce the truth tables of half adders. 43) List the output expressions for the Sum and Carry of the half adder 44) Draw the circuit of the half-adder. 45) Define full adder. 46) Produce the truth table for the full adder. 47) List the output expressions for the Sum and Carry of the full adder. 48) Draw the full adder circuit using the standard design having two half-adders and one OR gate.

**Subject Focus:****Computer Architecture and Assembly Language****Learning Outcome 2:**

I can explain how different hardware components interoperate in a computer system and how these can be programmed using Assembly.

Topic	Sub-Topic	Assessment Criteria
2.1 Overview of the Organisation of a Computer System	2.1.1 Main components	1) Outline the main components of a computer. <i>Limited to the Central Processing Unit (CPU), Graphics Processing Unit (GPU), registers, main memory, auxiliary storage, input devices, output devices, and system bus.</i> 2) Draw a simple Von Neumann architecture model. <i>Limited to the CPU, main memory, auxiliary storage, input devices, output devices and system bus.</i>
	2.2 Components of a Computer System	3) Define the system bus. 4) Define the buses included in the system bus. <i>Limited to the address bus, data bus, and control bus.</i> 5) Define word length / size. 6) Define addressable space. 7) Outline the relationship between the size of the word and the size of the data bus. 8) Calculate the addressable space and the total memory capacity of a device using index form. 9) Define the system clock. 10) Distinguish between synchronous and asynchronous data transfer. 11) Outline the steps of the memory read cycle. 12) Outline the steps of the memory write cycle.
	2.2.2 Memory	13) Compare and contrast the characteristics of Dynamic RAM (DRAM) and Static RAM (SRAM). 14) Give examples of applications for DRAM and SRAM for a given scenario. 15) Compare and contrast the characteristics of ROM type memory chips. <i>Limited to ROM and Electrically Erasable Programmable ROM (EEPROM).</i>

Topic	Sub-Topic	Assessment Criteria
		<p>16) Give an example of the application of each type of ROM chip mentioned in section 2.2.2 (15) for a given scenario.</p>
	2.2.3 CPU	<p>17) Draw a block diagram of the CPU. <i>Limited to the Program Counter (PC), Memory Address Register (MAR), Memory Buffer Register (MBR)/ Memory Data Register (MDR), Instruction Register (IR), Control Unit (CU), Arithmetic and Logic Unit (ALU), system bus, and cache memory.</i></p> <p>18) Outline the function of the CU.</p> <p>19) Outline the function of the ALU.</p> <p>20) List the steps in the fetch, decode, and execute cycles.</p> <p>21) Outline the main low-level operations involved in the steps of the fetch, decode and execute cycle. <i>Limited to instructions of mathematical operations SUB and ADD and the data transfer operation MOV to and from registers as well as main memory.</i></p> <p>22) Explain the use of stack structure during process execution. <i>Limited to the stack's role in subroutine transfer.</i></p> <p>23) Define an instruction set.</p>
	2.2.4 On-chip CPU storage	<p>24) Outline the purpose and use of the following registers in the functioning of the CPU. <i>Limited to:</i></p> <ul style="list-style-type: none"> <li>● <i>Stack registers;</i></li> <li>● <i>PC;</i></li> <li>● <i>MAR;</i></li> <li>● <i>MBR/ MDR;</i></li> <li>● <i>IR;</i></li> <li>● <i>Accumulator (AX);</i></li> </ul>

Topic	Sub-Topic	Assessment Criteria
		<ul style="list-style-type: none"> <li>● <i>Base Register (BX);</i></li> <li>● <i>Count Register (CX);</i></li> <li>● <i>Data Register (DX).</i></li> </ul> <p>25) Outline the purpose and use of CPU caches. <i>Limited to L1 and L2 caches.</i></p> <p>26) Compare and contrast cache, registers, and RAM. <i>In terms of access speeds and system performance.</i></p>
	2.2.5 I/O peripherals	<p>27) Outline the basic functionality of I/O related technologies. <i>Limited to:</i></p> <ul style="list-style-type: none"> <li>● <i>Serial data transmission (e.g., USB Ports);</i></li> <li>● <i>Parallel data transmission (e.g., NVMe).</i></li> </ul>
2.3 Assembly languages	2.3.1 Representing an instruction set	<p>28) Classify the categories of Low-Level instructions in a CPU instruction set. <i>Categorised (in Appendices as: Data Transfer Instructions, Logical Instructions, Arithmetic Instructions, Transfer Instructions, Flag Manipulation, Shift and Rotate</i></p> <p>29) Outline what is a CPU-executable instruction. <i>In terms of operands and opcodes.</i></p> <p>30) Define mnemonics. <i>In terms of opcodes.</i></p> <p>31) Define a pseudo-directive. <i>Limited to halt and end.</i></p> <p>32) Interpret simple programs written in assembly.</p>

Topic	Sub-Topic	Assessment Criteria
		<p><i>Limited to the instruction set given in Appendices based on the instruction set of the 8086 processor.</i></p> <p><i>Each instruction has to be followed by a comment.</i></p> <p>33) Outline the addressing modes which can be used in assembly.</p> <p><i>Limited to:</i></p> <ul style="list-style-type: none"> <li>● <i>Register addressing, e.g. MOV AX, BX (move the contents of register BX into register AX);</i></li> <li>● <i>Immediate addressing, e.g. MOV AX, #03H (move value 3 hex into register AX);</i></li> <li>● <i>Direct addressing, e.g. MOV AX, 0810H (move the contents of memory location 0810H into the accumulator);</i></li> <li>● <i>Indirect addressing, e.g. MOV AX, [BX] (the contents of the BX register is an address and is used to point to the memory location where the data is to be found);</i></li> </ul> <p>34) Define symbolic addressing.</p>

**Subject Focus:****Operating Systems****Learning Outcome 3:****I can explain the functions of different operating systems, how they manage memory, handle file systems, and process input and output operations.**

Topic	Sub-Topic	Assessment Criteria
3.1 Introduction	3.1.1 Booting process	1) Describe the booting process of the BIOS.
	3.1.2 Software types	2) Describe the difference between system and application software.
	3.1.3 Translators	3) Describe the following translators: <ul style="list-style-type: none"> <li>• Assemblers;</li> <li>• Compilers;</li> <li>• Interpreters.</li> </ul>
3.2 Types of OS		4) Explain the functions of: <ul style="list-style-type: none"> <li>• Batch;</li> <li>• Mobile;</li> <li>• Real-Time;</li> <li>• Distributed.</li> </ul>
		5) Give examples of uses for each type of OS mentioned in part (4).
3.3 Process management	3.3.1 Process states	6) Define a process. 7) Explain the three process states: <ul style="list-style-type: none"> <li>• Ready;</li> <li>• Running;</li> <li>• Waiting/Blocked.</li> </ul>

Topic	Sub-Topic	Assessment Criteria
3.4. Scheduling	3.4.1 Scheduling algorithms	<p>8) Explain the need for scheduling.</p> <p>9) Differentiate between pre-emptive and non pre-emptive scheduling.</p> <p>10) Explain how processes can be scheduled using the Round Robin algorithm.</p>
3.5 Memory Management	3.5.1 Memory partitioning	<p>11) Distinguish between a logical and a physical address space.</p> <p>12) Explain the need for Memory partitioning.</p> <p>13) Outline how fixed memory allocation works.</p> <p>14) Outline how dynamic memory allocation works.</p> <p>15) Distinguish between internal and external fragmentation.</p>
3.6 File Management	3.6.1 File attributes	<p>16) List the file attributes:</p> <ul style="list-style-type: none"> <li>• Name;</li> <li>• Type;</li> <li>• Location;</li> <li>• Size;</li> <li>• Protection;</li> <li>• Time, date and user identification.</li> </ul>
	3.6.2 File protection	<p>17) Explain how files may be protected.</p> <p><i>Limited to:</i></p> <ul style="list-style-type: none"> <li>• <i>User ID and password;</i></li> <li>• <i>User Home Directory;</i></li> <li>• <i>File access rights and allocated privileges;</i></li> <li>• <i>File attributes;</i></li> <li>• <i>Backup in case of hardware failure.</i></li> </ul>

Topic	Sub-Topic	Assessment Criteria
3.7 Handling of I/O Operations	3.7.1 Interrupt handling	<p>18) Define an Interrupt. <i>Limited to hardware and software interrupts.</i></p> <p>19) Define an Interrupt Request (IRQ).</p> <p>20) Define the function of the Interrupt Handler / Interrupt Service Routine (ISR).</p> <p>21) Outline the use of the interrupt register and interrupt enable / disable register.</p> <p>22) Explain the difference between polling vs. vectored interrupt approach (including the use of the vector table).</p> <p>23) Explain how multiple interrupts are handled understanding the scheduling algorithm implementation of priority.</p>

**Subject Focus:****Networking****Learning Outcome 4:**

**I can develop a comprehensive understanding of networking concepts, technologies, and their practical applications in various contexts.**

Topic	Sub-Topic	Assessment Criteria
4.1 Basics of Networking	4.1.1 Benefits of Networks	1) Discuss the benefits of networking. 2) Describe the roles of computing devices in a network. <i>Limited to switch, router and server.</i>
	4.1.2 Network Topologies	3) Outline bus, star, mesh and hybrid topologies. 4) Compare and contrast the characteristics of a local area network (LAN) and a wide area network (WAN).
4.2 Connecting devices	4.2.1 Wired networks	5) Compare coaxial, twisted pair and fibre optic media. <i>Limited to relative speed, material, transmission technology and relative transmission distance.</i>
	4.2.2 Wireless networks	6) Compare Bluetooth, WiFi, cellular, and satellite media. <i>Limited to relative speed, data capacity (bandwidth) and relative transmission distance.</i> 7) Describe the hardware used to support a wireless network. <i>Limited to Wireless Network Interface Card (WNIC) and Wireless Access Point.</i> 8) Differentiate between the use of wired and wireless networks.
	4.2.3 Addressing	9) Outline the need of addressing in networks. 10) Define the IPv4 and IPv6 addressing formats. 11) Allocate IP host addresses given a network address. <i>Limited to IPv4.</i>

Topic	Sub-Topic	Assessment Criteria
		12) Outline the use of subnetting in a network. 13) Differentiate between a public IP address and a private IP address. 14) Differentiate between static IP address and dynamic IP addressing.
4.3 Protocols	4.3.1 Introduction to protocols	15) Define what a protocol is. 16) Outline why a protocol is needed to allow communication between devices. 17) Outline the TCP/IP protocol suite.
4.4 Cloud Computing		18) Define Cloud Computing. 19) Outline Software as a Service (SaaS). 20) Justify the use of cloud computing.

**Subject Focus:** Programming in Python

**Learning Outcome 5:** I can design, write, test, and debug Python programs using structured and object-oriented programming techniques to solve real-world problems effectively and efficiently.

Topic	Sub-Topic	Assessment Criteria
5.1 Input and output	5.1.1 Input and Output statements	<p>1) Interpret a program snippet that includes output statements. <i>Limited to print() function without output formatting.</i></p> <p>2) Interpret a program snippet that includes output and /or formatted output statements. <i>Limited to:</i></p> <ul style="list-style-type: none"> <li>• <i>new line &amp; no new line in print function;</i></li> <li>• <i>the use of the F-Strings or String Concatenation (String+Var).</i></li> </ul> <p>3) Develop a program using output statements according to a given scenario.</p> <p>4) Develop a program snippet that includes output and/or formatted output statements according to a given scenario.</p>
5.2 Variables	5.2.1 Handling Variables	<p>5) Identify between different types of variables for a given scenario. <i>Limited to: integer, float, String and Boolean.</i></p> <p>6) Interpret a program snippet that includes the use of variables and/or input statements. Program-snippets may include variable initialization. <i>Limited to:</i></p> <ul style="list-style-type: none"> <li>• <i>input() function;</i></li> <li>• <i>variables as listed in (5).</i></li> </ul> <p>7) Interpret a program snippet that includes the use of variables and/or type conversion functions and/or input statements. <i>Limited to int(), float(), and str() type conversions</i></p>

Topic	Sub-Topic	Assessment Criteria
		<p>8) Develop a program that includes the use of variables and/or input statements. <i>Limited to those listed in (6).</i></p> <p>9) Develop a program using variables and/or type conversion functions and/or input statements. <i>Limited to those listed in (7).</i></p>
5.3 Arithmetic Operations	5.3.1 Arithmetic Operations	<p>10) Identify arithmetic operations in a given scenario. <i>Limited to the following arithmetic operators: = equals, / division, + addition, // floor division, - subtraction, % modulus, * multiplication, ** power of</i></p> <p>11) Interpret a program snippet that includes arithmetic operations</p> <p>12) Interpret a program snippet that includes arithmetic assignment operators. <i>Limited to: +=, -=, *=, /=, //=, %=</i></p> <p>13) Develop a program using arithmetic operations. <i>Limited to those listed in (10).</i></p> <p>14) Develop a program using arithmetic assignment operators. <i>Limited to those listed in (12).</i></p>
5.4 Decision Statements	5.4.1 Handling Variables	<p>15) Interpret a program snippet that includes decision statements. <i>Limited to:</i></p> <ul style="list-style-type: none"> <li>▪ <i>Decision statements: if, elif, else</i></li> <li>▪ <i>Conditional operators: ==, !=, &gt;, &lt;, &gt;=, &lt;=</i></li> <li>▪ <i>Logical Operators: and, or, not</i></li> <li>▪ <i>Membership operators: in, not in</i></li> </ul>

Topic	Sub-Topic	Assessment Criteria
		<p>16) Interpret a program snippet that includes nested decision statements.</p> <p>17) Develop a program using decision statements.</p> <p><i>Limited to those listed on (15).</i></p> <p>18) Develop a program using nested decision statements.</p>
5.5 Iteration	5.5.1 Iteration statements	<p>19) Interpret a program snippet that includes iteration statements.</p> <p><i>Limited to: for-in loop, while loop, and while-else loop.</i></p> <p><i>Note: the for-in loop may include the range() function.</i></p> <p>20) Interpret a program snippet that includes nested iteration statements.</p> <p><i>Note: may include 'break' and 'continue' statements.</i></p> <p>21) Develop a program using iteration statements.</p> <p><i>Limited to those in (19).</i></p> <p>22) Develop a program using nested iteration statements.</p>
5.6 Functions	5.6.1 Using and defining functions	<p>23) Define the term function and/or module.</p> <p>24) Distinguish between function and module.</p> <p>25) Distinguish between built-in function and user defined function.</p> <p>26) Interpret a program snippet that includes String functions.</p> <p><i>Limited to: Get character from string (String[position]), Substring (String[2:5]), strip(), len(), lower(), upper(), replace(), find(), count().</i></p> <p>27) Interpret a program snippet that includes user defined functions.</p> <p>28) Develop a program using String functions.</p>

Topic	Sub-Topic	Assessment Criteria
		<p>29) Develop a program using user-defined functions.</p> <p><i>Note: user-defined functions may include value return and parameters.</i></p>
5.7 Data Structures	5.7.1 Lists and Dictionaries	<p>30) Interpret a program snippet that includes lists as data structures.</p> <p><i>Limited to the use of the following functions: append(), clear(), count(), insert(), len(), pop(), remove(), reverse(), and sort().</i></p> <p>31) Interpret a program snippet that includes dictionaries as data structure.</p> <p><i>Limited to the use of the following functions: clear(), items(), get(), and pop().</i></p> <p>32) Develop a program using lists as data structures.</p> <p><i>Limited to those in (30).</i></p> <p>33) Develop a program using dictionaries as data structures.</p> <p><i>Limited to those in (31).</i></p>
5.8 Modules	5.8.1 Random module	<p>34) Interpret the statement import and/or from import and/or from-import-as.</p> <p><i>Examples: ▪ import random ▪ from random import randint ▪ from random import * ▪ from random import randrange as num_generator</i></p> <p>35) Interpret a program snippet that includes the Random module.</p> <p><i>Limited to randint(start, stop) function.</i></p> <p>36) Interpret a program snippet that includes the Random module to access and modify items in data structures.</p> <p><i>Limited to: choice(), shuffle() and sample() from a List.</i></p> <p>37) Develop a program using the Random module.</p> <p><i>Limited to those in (35).</i></p>

Topic	Sub-Topic	Assessment Criteria
		38) Develop a program using the Random module to access and/or modify items in data structures. <i>Limited to those in (36).</i>
5.9 Programming Errors and Exceptions	5.9.1 Types of programming errors  5.9.2 Handling exceptions  5.9.3 Debugging	39) Distinguish between syntax, logical and runtime errors. 40) Explain the effect of an error on program compilation and/or runtime. <i>Limited to:</i> <ul style="list-style-type: none"> <li>• <i>syntax: does not compile;</i></li> <li>• <i>logical: gives undesired result;</i></li> <li>• <i>runtime: stops executing.</i></li> </ul> 41) Explain the use of a 'try..except..finally' block. 42) Interpret a program using exception handling. <i>Limited to ZeroDivisionError, FileNotFoundError, ValueError</i> 43) Develop a program using (42). 44) Outline commonly used debugging tools. <i>Limited to logs, breakpoints and stepping, and variable inspection.</i>
5.10 Object Oriented Programming (OOP)	5.10.1 Classes and Instances	45) Outline the structure of a class. <i>In terms of attributes and methods.</i> 46) Outline the concept of constructor methods. 47) Define the 'self' keyword. 48) Relate classes and objects/instances. 49) Interpret a program using (45) to (48).

Topic	Sub-Topic	Assessment Criteria
		50)Develop a program using 45) to 48).
	5.10.2 Encapsulation	51)Outline the concept of encapsulation. 52)Interpret a program using private attributes and public get( ) and set( ) methods. 53)Develop a program using private attributes and public get( ) and set( ) methods.

**Subject Focus:** Systems Analysis and Design

**Learning Outcome 6:** I can analyse and design systems using established methodologies.

Topic	Sub-Topic	Assessment Criteria
6.1 Overview of the Systems Development Life Cycle (SDLC)	6.1.1 Introduction to SDLC	1) List the stages of the Systems Development Life Cycle (SDLC). <i>Limited to Problem Definition, Feasibility Study, Requirements Elicitation, Analysis, Design, Coding and Testing, Implementation, Deployment, Maintenance, Retirement.</i>
	6.1.2 Software Development Methodologies	2) Justify the need of the SDLC structure. <i>Limited to the need for analysis, the idea of scope creep, and the need to have a structure for developing software.</i>
6.2 The Waterfall Life Cycle software development model	6.2.1 The stages of the Waterfall Life Cycle	3) Define the Waterfall Life Cycle. 4) List the stages of the Waterfall Life Cycle. <i>Limited to Problem Definition and Feasibility Study, Requirements Analysis, System Design, Implementation, Verification/Testing, Deployment, and Maintenance.</i> 5) Define the stages of the Waterfall Life Cycle (the stages listed in section 6.2.1 (4)). 6) List the advantages and disadvantages of using the Waterfall Life Cycle.
	6.2.2 The role of the Systems Analyst in the Waterfall Life Cycle	7) List the roles of the Systems Analyst.
6.3 Problem identification		8) State what prompts an organisation to develop a new system.

Topic	Sub-Topic	Assessment Criteria
6.4 Feasibility Study		<p>9) Define Feasibility Study.</p> <p>10)List the aspects of the feasibility study.</p> <p><i>Limited to:</i></p> <ol style="list-style-type: none"> <li>1. <i>technical;</i></li> <li>2. <i>operational;</i></li> <li>3. <i>timeliness and relevance;</i></li> <li>4. <i>economic;</i></li> <li>5. <i>legal;</i></li> <li>6. <i>social.</i></li> </ol>
6.5 Requirements Elicitation	6.5.1 Techniques used to understand a problem completely.	<p>11)List the fact-finding techniques used to understand the problem completely.</p> <p><i>Limited to:</i></p> <ol style="list-style-type: none"> <li>1. <i>interviews;</i></li> <li>2. <i>questionnaires;</i></li> <li>3. <i>inspection of documents;</i></li> <li>4. <i>observation (of existing systems and work processes).</i></li> </ol> <p>12)Recommend fact-finding technique/s for a given scenario.</p>
	6.5.2 Off-the-shelf and Purpose-built software development	<p>13)Differentiate between Off-the-shelf and Purpose-built software development.</p> <p>14)Justify the choice of software that can be used in a given scenario.</p>
	6.5.3 Open source and proprietary owned solutions.	<p>15)Differentiate between Open source and Proprietary owned solutions.</p> <p>16)Evaluate specific scenarios to determine which type of coding solutions in (19) should be employed.</p>
6.6 Design techniques	6.6.1 Introduction to Design techniques	<p>17)Define an algorithm.</p> <p>18)Distinguish between top-down and bottom-up approach when designing software.</p>

Topic	Sub-Topic	Assessment Criteria
		19) Define techniques which could be used to design a system. <i>Limited to flowcharts and pseudocodes.</i>
	6.6.2 Modularity	20) Define modularity/modular programming. 21) List the advantages of modular programming/modularity. 22) List the disadvantages of not developing software in a modular way.
6.7 Development and Testing	6.7.1 Development	23) Justify the importance of adhering to best practices and following coding standards.
	6.7.2 Testing Strategies	24) Identify appropriate test data for a given scenario. <i>Limited to testing using valid; invalid and extreme data.</i> 25) Distinguish between the following testing strategies: <ul style="list-style-type: none"> <li>• Black-box and white-box testing.</li> <li>• Alpha and Beta testing.</li> </ul>
6.8 Documentation		26) Explain the importance of documentation.
6.9 Deployment	6.9.1 Deployment phase	27) Describe what happens during the deployment phase. <i>Limited to users' education and training, installation of additional software and hardware, and the creation of user files.</i>
	6.9.2 Changeover techniques	28) List the main changeover techniques: <ol style="list-style-type: none"> <li>1. Parallel;</li> <li>2. Phased;</li> <li>3. Direct;</li> <li>4. Pilot.</li> </ol> 29) Compare and contrast between the changeover techniques as in (32)

Topic	Sub-Topic	Assessment Criteria
6.10 Maintenance	6.10.1 Introduction to maintenance	30) Justify a suitable changeover technique for a given scenario. 31) Discuss the importance of maintenance of software systems. 32) List the types of maintenance <ol style="list-style-type: none"> <li>1. Adaptive;</li> <li>2. Corrective;</li> <li>3. Perfective;</li> <li>4. Preventive.</li> </ol>
6.11 Ethics		33) Identify the need for ethics as a computing professional. 34) Reflect on the need to act ethically and its impact in a given situation. 35) Explain the need for copyright legislation.

<b>Subject Focus:</b>	<b>Databases</b>
<b>Learning Outcome 7:</b>	<b>I can understand foundational database concepts, including structures, DBMS role, three-level architecture, relational model, E-R modelling, normalisation, and basic SQL.</b>

Topic	Sub-Topic	Assessment Criteria
7.1 Introduction to Databases and Database Management Systems (DBMS)	7.1.1 Introduction to databases	1) Define a database. 2) Define a flat file database. 3) Identify limitations of a flat file database.
	7.1.2 Introduction to DBMS	4) Define a DBMS. 5) Describe the structure of a DBMS. <i>To include schema, data dictionary, and database languages (Data manipulation language (DML) and Data description language (DDL).</i> 6) Differentiate between DML and DDL. 7) Describe the functions of a DBMS. <i>Limited to data storage retrieval and update, backup and recovery services, integrity services, provides an interface, and security services.</i>
7.2 Database Administrator (DBA)		8) Define the role of the DBA. 9) List the responsibilities of the DBA.
7.3 Relational Databases		10) Define a relational database. 11) Define the basic terms: <ul style="list-style-type: none"> <li>• record/tuple;</li> <li>• field/attribute;</li> <li>• primary key;</li> </ul>

Topic	Sub-Topic	Assessment Criteria
<p>7.4 Entity-Relationship (E-R) Diagrams</p> <p><i>The standard Crow's Foot notation is to be used (Appendices)</i></p>		<ul style="list-style-type: none"> <li>• secondary key;</li> <li>• foreign key;</li> <li>• composite key;</li> <li>• entity.</li> </ul> <p>12) Describe the logical structure of a relational database as a set of tables linked together using common fields.</p> <p>13) Describe the different types of cardinality.</p> <p><i>Limited to one-to-one, one-to-many, many-to-many.</i></p> <p>14) Interpret an E-R Diagram.</p> <p><i>Limited to three entities.</i></p> <p>15) Draw E-R Diagram from a given scenario.</p> <p><i>Limited to three entities.</i></p>
7.5 Normalisation		<p>16) Define Normalisation.</p> <p>17) Explain the need for normalisation.</p>
7.6 Structured Query Language (SQL)		<p>18) Define Query language.</p> <p>19) Interpret SQL commands.</p> <p><i>Limited to:</i></p> <p>SELECT... FROM...;</p> <p>SELECT... FROM...WHERE;</p> <p>SELECT... FROM...WHERE...AND/OR/NOT;</p> <p>SELECT... FROM...WHERE...ORDER BY... (using ascending or descending order).</p>

**Subject Focus:** Algorithms and Artificial Intelligence

**Learning Outcome 8:** I can represent algorithms using flowcharts and pseudocode, implement searching and sorting algorithms, and explain the technology behind artificial intelligence.

Topic	Sub-Topic	Assessment Criteria
8.1 Problem-solving and Algorithm Development	8.1.1 Computational Thinking	1) Define computational thinking. <i>In terms of the four competencies of decomposition, pattern recognition, abstraction, algorithmic thinking.</i> 2) Outline the benefits of computational thinking.
	8.1.2 Algorithms Basics	3) Outline the importance of algorithms in the design of solutions. 4) Represent algorithms using a flowchart and pseudocode. 5) Write pseudocode that contains an input, some process, and an output. 6) Write pseudocode that includes constructs of sequence, selection and repetition. 7) Draw a flowchart from pseudocode. 8) Draw a flowchart from a problem description. 9) Express an algorithm to the level of detail required such that the task can be programmed.
	8.1.3 Algorithms	10) Write pseudocode to implement linear search. 11) Write pseudocode to implement a binary search. 12) Interpret a bubble sort algorithm. 13) Describe the following abstract data types: single linked list and stack.
8.2 Artificial Intelligence (AI)		14) Define AI. 15) Argue the use of AI in an application.

Topic	Sub-Topic	Assessment Criteria
		16)Outline different forms of AI. <i>Limited to generative AI and natural language processing (NLP).</i> 17)Outline the ethical implications of AI on society and the economy. <i>Limited to the EU Commission AI Act (April 2024).</i>

## Scheme of Assessment

This subject is assessed by means of **TWO** components:

1. Written examination paper.
2. Portfolio.

Component	Section	Learning Outcome	% Weighting
Written Examination Paper	Section A	LO 5	15%
	Section B	All LOs	50%
	Section C	All LOs excluding LO 5	20%
Portfolio	Refer to notes hereunder	LO 5	15%

### Written Examination Paper

- candidates will write their answers on a separate booklet provided by MATSEC;
- includes three sections with a total of 85% of total score.

#### Section A

- carries 15% of the total score;
- consists of one compulsory question focused on LO 5;

#### Section B

- carries 50% of the total score;
- consists of eight compulsory short questions requiring to the point short answers.

#### Section C

- carries 20% of the total score;
- consists of two more in-depth questions of which the candidates will be required to answer one.

## Portfolio

The Portfolio carries a total of 100 marks amounting to 15% of the total score. Candidates should follow the structure hereunder. Award a maximum of half the marks in each section, if work presented does not meet all the mentioned criteria respectively.

Candidates are expected to provide **ONE** program for **each** section as explained hereunder. The portfolio is to be submitted according to directions set by MATSEC.

### General Guidelines (5%)

- Documentation of the code should include clear comments for readability and maintainability.
- Meaningful names should be used: variable names, user-defined functions, and class names.

### Section 1: Input, Output, and Variables (10%)

#### Input/Output: (5%)

- The print() function for output.
- The input() function for entering data.
- Formatting output using F-Strings, and string concatenation.

#### Variables: (5%)

- Use of integer, float, string, and boolean variable types.
- Inputting and storing data in variables.
- Use of type conversion functions (int(), float(), str()).

### Section 2: Arithmetic Operations and Decision Statements (15%)

#### Arithmetic Operations: (5%)

- Use of at least **THREE** from
  - = equals, / division, + addition, // floor division, - subtraction, % modulus, \* multiplication, \*\* power of.
  - Or included as +=, -=, \*=, /=, //=, and %=.

#### Decision Statements: (10%)

- Use of if, elif, and else statements along with conditional, logical, and membership operators.
- Use of at least **ONE** nested decision statement.

### Section 3: Iteration and Functions (30%)

#### Iteration: (15%)

- Use of iteration constructs, including at least **ONE** from for-in, while and while-else loops.
- Use of control statements such as break and continue within loops for flow control.

#### Functions: (15%)

- Use of at least **THREE** String functions from String functions from string (String[position]), Substring (String[2:5]), strip(), len(), lower(), upper(), replace(), find(), and count().
- Use of at least **TWO** user-defined functions with parameters and return values.

## Section 4: Data Structures and Modules (40%)

### Data Structures: (30%)

- Use of a list including at least the implementation of **THREE** functions (e.g., `append()`, `clear()`, `count()`, `insert()`, `len()`, `pop()`, `remove()`, `reverse()`, and `sort()`).

### Modules: (10%)

- Use of the `random` module and its associated functions.

### Note for Private candidates:

- Private Candidates are to submit the assignment to MATSEC for assessment, by the date stipulated by MATSEC. Candidates may be called for an interview about their work.
- An authentication form is required and is available for download from the MATSEC website.

## Appendices

### Assembly Languages

<i>Data Transfer instructions</i>	MOV	Moves byte or word to register or memory
	PUSH	Push a word on stack
	POP	Pop a word from stack
<i>Logical Instructions</i>	NOT	Logical not (1's complement)
	AND	Logical and
	OR	Logical or
	XOR	Logical exclusive-or
<i>Arithmetic Instruction</i>	ADD , ADC	Add and Add with carry
	SUB, SBB	Subtract and Subtract with borrow
	INC	Increment
	DEC	Decrement
	CMP	Compare
<i>Transfer Instructions</i>	JMP	Unconditional Jump
	JE	Jump on Equal
	JNE	Jump on Not Equal
	JL	Jump if Less
	JLE	Jump if less or equal
	JG	Jump if Greater
	JGE	Jump if Greater or Equal
	JC, JNC	Jump on carry or Jump on No Carry
	CALL	Call Subroutine
	RET	Return from subroutine
<i>Flag Manipulation</i>	CLC	Clear Carry
	STC	Set Carry
<i>Shift and Rotate</i>	SHL, SHR	Logical Shift Left or Right
	RCL, RCR	Rotate through Carry Left or Right

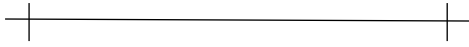
<i>Pseudo-directives</i>	HALT, END
--------------------------	-----------

**List of Acronyms**

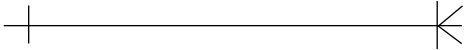
ADSL	Asymmetric Digital Subscriber Line
ASCII	American Standard Code for Information Interchange
ATM	Asynchronous Transfer Mode
BNF	Backus Naur Form
CISC	Complex Instruction Set Computer
CSMA/CD	Carrier Sense Multiple Access / Collision Detect
DMA	Direct Memory Access
DTP	Desktop Publishing
EBNF	Extended Backus Naur Form
ROM	Read Only Memory
EEPROM	Electrically Erasable Programmable ROM
EPROM	Erasable Programmable ROM
FDDI	Fiber Distributed Data Interface
FTP	File Transfer Protocol
HDSL	High bit-rate Digital Subscriber Line
IMAP	Internet Message Access Protocol
ISDN	Integrated Services Digital Network
LAN	Local Area Network
LIFO	List In First Out
MAN	Metropolitan Area Network
OSI	Open Systems Interconnection
POP	Post Office Protocol
PROM	Programmable ROM
RISC	Reduced Instruction Set Computers
SMTP	Simple Mail Transfer Protocol
USB	Universal Serial Bus
WAN	Wide Area Network

## Crow's Foot E-R Diagram Notation

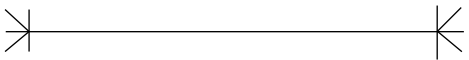
1 - 1



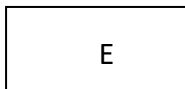
1 - M



M - N



## Data Entity



"E" is entity's name