L-Università ta' Malta

MATSEC
Examinations Board

AM 07 Syllabus

COMPUTING

2027

Table of Contents

Introduction

Computing is a dynamic field of study that has shaped modern society, with innovations that have revolutionized various industries. This syllabus is designed to provide candidates with a thorough understanding of the core principles and foundational concepts of computing, while also equipping them with essential technical knowledge to gain hands-on experience in the field. It aims to prepare candidates for further study or careers in computing and related areas.

The syllabus encourages candidates to explore the wide-ranging potential and challenges that computing presents to society, promoting critical thinking about both the opportunities and ethical issues associated with technological advancements. It builds on prior knowledge of computing, equivalent to what is taught at the Secondary Education Certificate level, and advances candidates' understanding through a structured approach.

Organized into Learning Outcomes (1 – 9), each with specific Assessment Criteria, the syllabus helps teachers, candidates, and examiners determine the scope and depth of the content to be covered. While lessons may go beyond these outcomes, assessments will focus on the material specified in the syllabus.

The course provides a balance of theoretical knowledge and practical skills, ensuring candidates are well-prepared for further academic pursuits in computing. Candidates will also develop a sense of responsibility in applying their computing skills, with particular emphasis on ethical considerations, data security, and the environmental impact of technology.

List of Subject Foci

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

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. represent algorithms using flowcharts and pseudocode, implement searching and sorting algorithms, and explain the technology behind artificial intelligence.

LO 4. explain the functions of different operating systems, how they manage memory, handle file systems, and process input and output operations.

LO 5. develop a comprehensive understanding of networking concepts, technologies, and their practical applications in various contexts.

LO 6. understand the fundamentals of formal languages, including syntax and semantics, parsing techniques and the compilation process.

LO 7. analyse and design systems using established methodologies.

LO 8. design, write, test, and debug Python programs using structured and object-oriented programming techniques to solve real-world problems effectively and efficiently.

LO 9. apply foundational database concepts, including structures, DBMS role, three-level architecture, relational model, E-R modelling, normalisation, and basic SQL.

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. *Limited to binary, decimal and hexadecimal.* 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 and Floating-point Representation	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. *Limited to 8 bits for the integer part and 4 bits for the fractional part.*
	1.3.2 Floating-point binary numbers	22)Represent signed floating-point binary numbers using two's complement. 23)Identify that the mantissa represents the fractional part and that the exponent represents the integer part. 24)Carry out conversions to and from decimal numbers. 25)Identify the range of signed floating point binary numbers using two's complement. *The number of bits for the mantissa should be limited to 8 bits and the number of bits for the exponent should be limited to 4 bits.*
	1.3.3 Normalisation of floating point numbers	26)Perform normalisation of floating-point positive and/or negative numbers using two's complement. 27)Carry out conversions to and from decimal numbers. 28)Identify the range of normalised signed floating point numbers using two's complement. *The number of bits for the mantissa should be limited to 8 bits and the number of bits for the exponent should be limited to 4 bits.*

Topic	Sub-Topic	Assessment Criteria
1.4 Errors	1.4.1 Errors in Computer Arithmetic	29)Define the arithmetic errors introduced due to register (space) size. *Limited to accuracy, precision, range, resolution, overflow, underflow, truncation error, rounding error and errors introduced when converting from decimal to binary.* 30)Carry out the calculation of such errors.
1.5 Character sets	1.5.1 Character Set Representation	31)Define the term 'character set'. 32)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). *Limited to 8 bits when using ASCII and 16 bits when using Unicode.* 33)Outline the difference between ASCII compared to UTF-8 as used in current systems.
1.6 Logic Gates	1.6.1 Basic Functions	34)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). 35)Define a logic gate and/or a logic circuit and/or a truth table. 36)Represent the AND, OR, NOT, NAND, NOR, XOR, and XNOR gates using standard symbols. 37)Draw the truth table for a given logic circuit and/or Boolean expression. *Limited to 4 inputs using the AND, OR, NOT, NAND, NOR, XOR, and XNOR gates.* 38)Derive the truth table to represent the solution for a given scenario. 39)Represent the AND, OR, NOT, NAND, NOR, XOR, and XNOR gates using Boolean expressions. 40)Draw the logic circuit for a given truth table and/or Boolean expression. *Limited to using AND, OR, NOT, NAND, NOR, XOR, and XNOR gates.* 41)Derive the logic circuit to represent the solution for a given scenario. *Limited to using AND, OR, NOT, NAND, NOR, XOR, and XNOR gates.*

Topic	Sub-Topic	Assessment Criteria
		42)Express the Boolean expression from a given logic circuit and/or truth table. *Limited to those listed in (34).* 43)Derive the Boolean expression to represent the solution for a given scenario. *Limited to those listed in (34).* 44)Complete a given logic circuit and/or truth table and/or Boolean expression. *Limited to those listed in (34).* 45)Interpret a given logic circuit and/or truth table and/or Boolean expression. *Limited to those listed in (34).*
1.7 Boolean Algebra	Laws of Boolean algebra	46)Outline the basic theorems and properties of the laws of Boolean algebra. *Limited to Appendices for Boolean Laws.* 47)Express a Boolean expression in its simplest form by applying the laws of Boolean algebra. *Limited to Appendices for Boolean Laws.* 48)Prove the laws of Boolean algebra and/or Boolean identities using truth tables. 49)Prove Boolean identities using truth tables and/or laws of Boolean algebra.
1.8 Karnaugh Maps (KM)	1.8.1 Simplification of Boolean expressions	50)Simplify two, three or four variable KM. *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 4 variable KM.*
	1.8.2 Application of KM	51)Interpret don't care conditions. 52)Use don’t care conditions to simplify KM.
	1.8.3 Half and Full Adder	53)Define half adder.

Topic	Sub-Topic	Assessment Criteria
		54)Define full adder. 55)Distinguish between half and full adder. 56)Produce the truth tables of half and full adders. 57)Represent the output expressions of the half and full adders using a KM. 58)Minimise using the laws of boolean algebra the output expressions produced for the half and full adders. 59)Draw the logic circuit diagram of the minimised half and a full adder.
	1.8.4 Magnitude comparator	60)Define a magnitude comparator. 61)Produce the truth table of a magnitude comparator using two, 2-bit binary numbers (4-inputs). 62)Represent the output expression/s of a magnitude comparator using a karnaugh map/s. 63)Minimise using the laws of boolean algebra the output expressions produced for the magnitude comparator. 64)Draw the logic circuit diagram of a minimised magnitude comparator.
	1.8.5 The 7-segment LED display	65)Define the 7-segment LED display. 66)Produce the truth table of the 7-segment LED display. 67)Represent the output expression/s of the 7-segment LED display using KM. *Limited to any three outputs.* 68)Minimise using the laws of boolean algebra the output expressions produced for the 7-segment LED display. *Limited to any three outputs.* 69)Draw the logic circuit diagram of a minimised.

Topic	Sub-Topic	Assessment Criteria
		Limited to any three outputs.
	1.8.6 Flip-Flops	70)Define a flip-flop. 71)Distinguish between JK and SR Flip-Flops. 72)Derive the truth table for the JK and SR Flip-Flops. 73)Draw the logic circuit diagram for the JK and SR Flip-Flops.
1.9 Universal Gates	1.9.1 NAND and NOR gates	74)Alter any logic expression to a NAND or NOR representation. 75)Draw any logic expression using NAND or NOR gates only.

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. *Limited to the Central Processing Unit (CPU), Graphics Processing Unit (GPU), registers, main memory, auxiliary storage, input devices, output devices, and system bus.* 2) Draw a simple Von Neumann architecture model. *Limited to the CPU, main memory, auxiliary storage, input devices, output devices and system bus*.
2.2 Components of a Computer System	2.2.1 The system bus	3) Define the system bus. 4) Define the buses included in the system bus. *Limited to the address bus, data bus, and control bus.* 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.

Topic	Sub-Topic	Assessment Criteria
	2.2.2 Memory	13)Outline the concept of address decoding. 14)Draw a block diagram of an address decoder. *Limited to 3 chip select lines and 7 address lines.* 15)Compare and contrast the characteristics of Dynamic RAM (DRAM) and Static RAM (SRAM). 16)Give examples of applications for DRAM and SRAM for a given scenario. 17)Compare and contrast the characteristics of ROM type memory chips. *Limited to ROM, Programmable ROM (PROM), Erasable PROM (EPROM), Electrically EPROM (EEPROM).* 18)Give an example of the application of each type of ROM chip mentioned in (17) for a given scenario. 19)Outline the concept behind the organisation of addresses in memory that shows how the memory is organised at the hardware level.
	2.2.3 CPU	20)Draw a block diagram of the CPU. *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.* 21)Outline the function of the CU. 22)Outline the function of the ALU. 23)List the steps in the fetch, decode, and execute cycle.

Topic	Sub-Topic	Assessment Criteria
		24)Outline the main low level operations involved in the steps of the fetch, decode and execute cycle. *Limited to instructions of mathematical operations SUB and ADD and the data transfer operation MOV to and from registers as well as main memory.* 25)Explain the use of stack structure during process execution. *Limited to the stack's role in subroutine transfer.* 26)Define an instruction set. 27)Compare the instruction sets of RISC-based and CISC-based architectures. 28)Define parallel processing. 29)Define pipelining. 30)Compare and contrast parallel processing and pipelining.
	2.2.4 On-chip CPU storage	31)Outline the purpose and use of the following registers in the functioning of the CPU. *Limited to:* ● *Index registers;* ● *Stack registers;* ● *Control registers;* ● *Status/Flag registers;* ● *PC* ● *MAR* ● *MBR/ MDR* ● *IR* ● *Accumulator (AX)* ● *Base Register (BX)* ● *Count Register (CX)* ● *Data Register (DX)*

Topic	Sub-Topic	Assessment Criteria
		32)Outline the purpose and use of CPU caches. *Limited to L1 and L2 caches.* 33)Compare and contrast cache, registers, and RAM. *In terms of access speeds and system performance.*
	2.2.5 I/O peripherals	34)Outline the basic functionality of I/O related technologies. *Limited to Serial data transmission (e.g., USB Ports) and Parallel data transmission (e.g., NVMe).*
2.3 Assembly languages	2.3.1 Representing an instruction set	35)Classify the categories of Low-Level instructions in a CPU instruction set. *Categorised (in Appendices) as: Data Transfer Instructions, Logical Instructions, Arithmetic Instructions, Transfer Instructions, Flag Manipulation, Shift and Rotate* 36)Outline what is a CPU-executable instruction. *In terms of operands and opcodes.* 37)Define mnemonics. *In terms of opcodes.* 38)Define a pseudo-directive. *Limited to halt and end.* 39)Interpret simple programs written in assembly. *Limited to the instruction set given in Appendices based on the instruction set of the 8086 processor.* *Each instruction has to be followed by a comment.* 40)Outline the addressing modes which can be used in assembly.

Topic	Sub-Topic	Assessment Criteria
		Limited to: ● *Register addressing, e.g. MOV AX, BX (move the contents of register BX into register AX);* ● *Immediate addressing, e.g. MOV AX, #03H (move value 3 hex into register AX);* ● *Direct addressing, e.g. MOV AX, 0810H (move the contents of memory location 0810H into the accumulator);* ● *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);* ● *Indexed addressing, e.g. MOV CX, [BX + DI] (the value in the base index register BX is combined with the number in the destination index register Direct Index or Source Index to provide the address of the number to be loaded into the CX register);* 41)Define symbolic and absolute addressing. 42)Outline the stages in the assembly process. *Limited to assembling, linking, loading, and relocation.* 43)Outline the purpose of a cross assembler.

Subject Focus:	**Algorithms and Artificial Intelligence**
Learning Outcome 3:	**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
3.1 Problem-solving and Algorithm Development	3.1.1 Computational Thinking	1) Define computational thinking. *In terms of the four competencies of decomposition, pattern recognition, abstraction, algorithmic thinking.* 2) Outline the benefits of computational thinking.
	3.1.2 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.
	3.1.3 Algorithms	10) Write an algorithm to implement linear search. 11) Write an algorithm to implement a binary search. 12) Argue how the performance of binary search is affected by the number of data items. 13) Write an algorithm to implement an insertion sort. 14) Write an algorithm to implement a bubble sort. 15) Describe how the performance of sorting algorithms may be affected by the initial order of the data and the number of data items.

Topic	Sub-Topic	Assessment Criteria
		16)Describe the following abstract data types: linear/circular queue, single linked list, binary tree, and stack. 17)Write algorithms to insert/delete an item in/from a linear/circular queue, single linked list and stack. 18)Write the steps involved to traverse a binary tree using pre-order, in-order, or post-order. 19)Describe the features of a directed graph. 20)Justify the use of a directed graph for a given scenario. 21)Interpret a directed graph for a given scenario. 22)Define recursion. 23)Describe the main features of the recursive process. *Limited to base and general case.* 24)Interpret a recursive algorithm for a given problem. 25)List the advantages and disadvantages of recursion.
3.2 Artificial Intelligence (AI)		26)Define AI. 27)Argue the use of AI in an application. 28)Outline different forms of AI. *Limited to: artificial neural networks (ANN), machine learning (ML), generative AI and natural language processing (NLP).* 29)Explain how artificial neural networks can be used for ML. 30)Compare supervised and unsupervised learning. 31)Describe the process of backpropagation of errors and regression in machine learning algorithms.

Topic	Sub-Topic	Assessment Criteria
		32)Outline the reasons for using ML and Deep Learning. 33)Discuss the ethical implications of AI on society and the economy. *Limited to the EU Commission AI Act (April 2024).*

Subject Focus:	**Operating Systems**
Learning Outcome 4:	**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
4.1 Introduction	4.1.1 Booting process	1) Outline the six steps in the booting process being: ● BIOS and Setup Program; ● The Power-On-Self-Test (POST); ● Loading the Operating System (OS); ● System Configuration; ● System Utility Loads; ● User Authentication.
4.2 Types of OS		2) Explain the functions of: ● Batch; ● Mobile; ● Real-Time; ● Distributed. 3) Give examples of uses for each type of OS mentioned in (2).
4.3 Process management	4.3.1 Process states	4) Define a process. 5) Explain the three process states: ● Ready; ● Running; ● Waiting/Blocked.
4.4. Scheduling	4.4.1 Scheduling algorithms	6) Explain the need for scheduling. 7) Differentiate between pre-emptive and non-pre-emptive scheduling.

Topic	Sub-Topic	Assessment Criteria
		8) Explain how processes can be scheduled using the following algorithms: ● Round Robin; ● Priority; ● First Come, First Served (FCFS).
4.5 Deadlocks	4.5.1 Defining deadlocks	9) Define deadlock. 10) Explain each of the four conditions for a deadlock to occur: ● mutual exclusion; ● hold and wait; ● no pre-emption; ● circular wait.
	4.5.2 Handling deadlocks	11) Outline deadlock prevention. 12) Outline deadlock detection. 13) Outline deadlock recovery in terms of: ● forced pre-emption of resources; ● forced termination of lower priority process; ● roll-back.
4.6 Memory Management	4.6.1 Memory partitioning	14) Distinguish between a logical and a physical address space. 15) Explain the need for Memory partitioning. 16) Outline how fixed memory allocation works. 17) Outline how dynamic memory allocation works. 18) Compare and contrast between fixed and dynamic memory partitioning systems. 19) Distinguish between internal and external fragmentation.

Topic	Sub-Topic	Assessment Criteria
	4.6.2 Paging	20)Outline the use of compaction to resolve external fragmentation. 21)Outline how the paging mechanism works. 22)Explain the concept of virtual memory. 23)Differentiate between pages and page frames. 24)Outline how a page table is used in this context. 25)Evaluate considerations for different page sizes.
4.7 File Management	4.7.1 File attributes and file operations	26)List the file attributes: ● Name; ● Type; ● Location; ● Size; ● Protection; ● Time, date and user identification. 27)List the file operations: ● Create; ● Write; ● Copy; ● Read; ● Move; ● Delete; ● Truncate.
	4.7.2 File Allocation Storage Techniques	28)Explain the following file allocation storage techniques: ● Contiguous; ● Linked; ● Indexed.

Topic	Sub-Topic	Assessment Criteria
		29)Compare and contrast the meaning and structure of file blocks in terms of: ● Contiguous; ● Linked; ● Indexed.
	4.7.3 File access methods	30)Explain the file access methods: ● Serial; ● Sequential; ● Random; ● Indexed Sequential. 31)Differentiate between the file access methods: ● Serial; ● Sequential; ● Random; ● Indexed Sequential.
	4.7.4 File protection	32)Explain how files may be protected. *Limited to:* ● *User ID and password;* ● *User Home Directory;* ● *File access rights and allocated privileges;* ● *File attributes;* ● *Backup in case of hardware failure.*
4.8 Handling of I/O Operations	4.8.1 Device communication	33)Explain the concept of device handshaking.

Topic	Sub-Topic	Assessment Criteria
	4.8.2 Memory mapped vs. Isolated I/O	34) Explain the following I/O Addressing techniques: ● Memory mapped I/O; ● Isolated I/O. 35) Differentiate between Memory mapped and Isolated I/O.
	4.8.3 Interrupt handling	36) Define an Interrupt. *Limited to hardware and software interrupts.* 37) Define an Interrupt Request (IRQ). 38) Define the function of the Interrupt Handler / Interrupt Service Routine (ISR). 39) Outline the use of the interrupt register and interrupt enable/disable register. 40) Explain the difference between polling vs. vectored interrupt approach (including the use of the vector table). 41) Outline the use of the stack in interrupt handling. 42) Differentiate between maskable and non-maskable interrupts. 43) Explain how multiple interrupts are handled understanding the scheduling algorithm implementation of: ● Priority; ● First In First Out (FIFO). 44) Explain the use of the Direct Memory Access (DMA) in relation to interrupt transfer to RAM.

Subject Focus:	**Networking**
Learning Outcome 5:	**I can develop a comprehensive understanding of networking concepts, technologies, and their practical applications in various contexts.**

Topic	Sub-Topic	Assessment Criteria
5.1 Basics of Networking	5.1.1 Benefits of Networks	1) Discuss the benefits of networking. 2) Describe the roles of computing devices in a network. *Limited to switch, router and server.*
	5.1.2 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). 5) Compare and contrast client-server and peer-to-peer models of networked computers.
5.2 Connecting devices	5.2.1 Wired networks	6) Compare coaxial, twisted pair and fibre optic media. *Limited to relative speed, material, transmission technology and relative transmission distance.* 7) Describe the hardware used to support a wired network. *Limited to Network Interface Card (NIC), Switch, Server, Router, and cables.*
	5.2.2 Wireless networks	8) Compare Bluetooth, WiFi, cellular, and satellite media. *Limited to relative speed, data capacity (bandwidth) and relative transmission distance.* 9) Describe the hardware used to support a wireless network. *Limited to Wireless Network Interface Card (WNIC) and Wireless Access Point.* 10)Differentiate between the use of wired and wireless networks.

	5.2.3 Addressing	11)Outline the need of addressing in networks. 12)Define the IPv4 and IPv6 addressing formats. 13)Allocate IP host addresses given a network address. *Limited to IPv4.* 14)Outline the use of subnetting in a network. 15)Differentiate between a public IP address and a private IP address. 16)Differentiate between static IP address and dynamic IP addressing.
5.3 Protocols	5.3.1 Introduction to protocols 5.3.2 TCP/IP 5.3.3 Applications	17)Define what a protocol is. 18)Outline why a protocol is needed to allow communication between devices. 19)Outline the use of layers to represent protocol implementation. 20)Outline the TCP/IP protocol suite. 21)Outline the function of each layer in the TCP/IP protocol suite. 22)Define HTTP/S, FTP/S, POP3, and IMAP protocols. 23)Outline the purpose of HTTP, FTP, POP3, and IMAP protocols. 24)Outline how peer-to-peer protocols facilitate file sharing. *Example: BitTorrent*
5.4 Cloud Computing		25)Define Cloud Computing. 26)Outline Software as a Service (SaaS). 27)Justify the use of cloud computing.

Subject Focus:	**Language Translators**
Learning Outcome 6:	**I can understand the fundamentals of formal languages, including syntax and semantics, parsing techniques and the compilation process.**

Topic	Sub-Topic	Assessment Criteria
6.1 Formal languages and syntax definition	6.1.1 Formal languages	1) Define the terms “language syntax” and “language semantics.” 2) Compare and contrast between natural and formal languages. 3) Identify the need for semantics (meaning) as opposed to syntax (form).
	6.1.2 Syntax definition	4) Define a metalanguage. 5) Define the syntax of a language using the BNF notation. 6) Demonstrate the meaning and use of meta symbols and variables, using the follows: ::= defined as; < > non-terminal symbol; | selection. *Note: Terminal symbols should not be enclosed in single/ double quotes.* 7) Define the syntax of a language using syntax diagrams. 8) Distinguish between terminal and non-terminal symbols.
6.2 The syntax of a formal language	6.2.1 Parsing	9) Define top-down parsing and bottom-up parsing. 10) Draw a parse tree (top-down) to check that a language statement is syntactically correct according to a set of rules or productions. 11) Outline ambiguous parsing.
	6.2.2 Reverse Polish Notation (RPN)	12) Evaluate arithmetic statements using RPN. 13) Convert from post-fix to in-fix statements using a stack.

Topic	Sub-Topic	Assessment Criteria
		14)Convert from in-fix to post-fix statements using a binary tree. 15)Evaluate post-fix statements.
6.3 The compilation process		16)List the stages of the compilation process.
	6.3.2 Lexical Analysis	17)Describe lexical analysis in terms of: ● removing redundant text; ● handling simple errors; ● converting lexemes to tokens.
6.4 Language Translators	6.4.1 Syntax and Semantics Analysis	18)Describe syntax and semantics analysis in terms of: ● parsing sentences; ● creating and updating a symbol table; ● describing the process of detecting and handling compile-time errors.
	6.4.2 Code Optimisation and Generation	19)Describe code optimisation and generation in terms of: ● optimising code using simple techniques; ● describing this stage as the process of translation into object code; ● describing the linking process. 20)Apply optimisation techniques on a given snippet of code. *Limited to: loop unrolling, common subexpression elimination; dead code elimination.*
	6.4.3 Different Translators	21)Differentiate between assemblers, compilers and interpreters. 22)Recommend a language translator for a given scenario. 23)Justify the choice of language translator for a given scenario.

Subject Focus:	**Systems Analysis and Design**
Learning Outcome 7:	**I can analyse and design systems using established methodologies.**

Topic	Sub-Topic	Assessment Criteria
7.1 Overview of the Systems Development Life Cycle (SDLC)	7.1.1 Introduction to SDLC	1) List the stages of the Systems Development Life Cycle (SDLC). *Limited to Problem Definition, Feasibility Study, Requirements Elicitation, Analysis, Design, Coding and Testing, Implementation, Deployment, Maintenance, Retirement.* 2) Justify the need why the SDLC structure exists. *Limited to the need for analysis, the idea of scope creep, and the need to have a structure for developing software.*
	7.1.2 Software Development Methodologies	3) List other software development methodologies. *Limited to DSDM (Dynamic System Development Method), RAD (Rapid Application Development), and Agile (Scrum).*
7.2 The Waterfall Life Cycle software development model	7.2.1 The stages of the Waterfall Life Cycle	4) Define the Waterfall Life Cycle. 5) List the stages of the Waterfall Life Cycle. *Limited to Problem Definition and Feasibility Study, Requirements Analysis, System Design, Implementation, Verification/Testing, Deployment, and Maintenance.* 6) Define the stages of the Waterfall Life Cycle as in section 7.2.1 (5). 7) List the advantages and disadvantages of using the Waterfall Life Cycle. 8) List the roles of the Systems Analyst.
7.3 Agile Methodologies	7.3.1 Introduction to Agile Methodology	9) Define Agile methodology. 10)List the different types of Agile methodology. *Limited to Scrum and Kanban*

Topic	Sub-Topic	Assessment Criteria
		11)List the stages of Agile methodology: ● requirements; ● design; ● development; ● testing; ● deployment; ● review.
	7.3.2 Scrum Methodology	12)Outline how Scrum methodology works. *Including the concept of sprints.* 13)Differentiate between a Product Owner, Product Team, and Scrum Master. 14)Distinguish between Waterfall Life Cycle and Agile methodology. 15)Distinguish between Scrum and Agile methodology. 16)Recommend the most suitable methodology for a given scenario.
7.4 Problem identification		17)State what prompts an organisation to develop a new system.
7.5 Feasibility Study		18)Define Feasibility Study. 19)List the aspects of the feasibility study. *Limited to:* ● *technical;* ● *operational;* ● *timeliness and relevance;* ● *economic;* ● *legal;* ● *social.*

Topic	Sub-Topic	Assessment Criteria
		20)Define the aspects of the feasibility study as in (19).
7.6 Requirements Elicitation	7.6.1 Techniques used to understand a problem completely.	21)List the fact-finding techniques used to understand the problem completely. *Limited to:* ● *interviews;* ● *questionnaires;* ● *inspection of documents;* ● *observation (of existing systems and work processes).* 22)Recommend fact-finding technique/s for a given scenario.
	7.6.2 Off-the-shelf and Purpose-built software development	23)Differentiate between Off-the-shelf and Purpose-built software development. 24)Justify the choice of software that can be used in a given scenario.
	7.6.3 Open source and proprietary owned solutions.	25)Differentiate between Open source and Proprietary owned solutions. 26)Evaluate specific scenarios to determine which type of coding solutions in (25) should be employed.
7.7 Analysis techniques	7.7.1 Unified Modelling Language (UML)	27)Define Unified Modelling Language (UML) 28)Define a Use-Case Diagram (UCD). 29)Explain how a UCD helps in determining system requirements. 30)Interpret a Use-Case Diagram in a given scenario. *Limited to primary and secondary actors, use-cases, association, system boundary, and <<extends>> and <<includes>> stereotypes.* *Limited to a maximum of 8 use-cases.* 31)Construct a Use-Case Diagram for a given scenario within the limitations in (30).

Topic	Sub-Topic	Assessment Criteria
		32)Define a UML Class Diagram. 33)Interpret a UML class diagram in a given scenario. *Limited to five classes, visibility, and the use of association and generalisation.* 34)Construct a UML class diagram for a given scenario within the limitations in (33).
7.8 Design techniques	7.8.1 Introduction to Design techniques	35)Define an algorithm. 36)Distinguish between top-down and bottom-up approach when designing software. 37)Define techniques which could be used to design a system. *Limited to flowcharts and pseudocodes.*
	7.8.2 Modularity	38)Define modularity/modular programming. 39)List the advantages of modular programming/modularity. 40)List the disadvantages of not developing software in a modular way.
7.9 Development and Testing	7.9.1 Development 7.9.2 Testing Strategies	41) Justify the importance of adhering to best practices and following coding standards. 42)Identify appropriate test data for a given scenario. *Limited to testing using valid; invalid and extreme data.* 43)Distinguish between the following testing strategies: • Bottom-up and top-down testing; • Black-box and white-box testing; • Alpha and Beta testing.
7.10 Documentation		44)Explain the importance of documentation.
7.11 Deployment	7.11.1 Deployment phase	45)Describe what happens during the deployment phase.

Topic	Sub-Topic	Assessment Criteria
		Limited to users' education and training, installation of additional software and hardware, and the creation of user files.
	7.11.2 Changeover techniques	46)List the main changeover techniques: • Parallel; • Phased; • Direct; • Pilot. 47)Compare and contrast between the changeover techniques as in (46) 48)Justify a suitable changeover technique for a given scenario.
7.12 Maintenance	7.12.1 Introduction to maintenance	49)Discuss the importance of maintenance of software systems. 50)List the types of maintenance: • Adaptive; • Corrective; • Perfective; • Preventive. 51)Distinguish between the types of maintenance as in (50). 52)Justify a type of maintenance for a given scenario.
7.13 Ethics		53)Identify the need for ethics as a computing professional. 54)Reflect on the need to act ethically and its impact in a given situation. 55)Explain the need for copyright legislation. 56)Contrast different types of software licensing. *Limited to Open source, Shareware, Public Domain, and Proprietary.* 57)Justify the use of a software license, as listed in (56), for a given scenario.

Subject Focus:	**Programming in Python**
Learning Outcome 8:	**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
8.1 Input and output	8.1.1 Input and Output statements	1) Interpret a program snippet that includes output statements. *Limited to print() function without output formatting.* 2) Interpret a program snippet that includes output and /or formatted output statements. *Limited to:* • *new line & no new line in print function;* • *the use of the F-Strings or String Concatenation (String+Var).* 3) Develop a program using output statements according to a given scenario. 4) Develop a program snippet that includes output and/or formatted output statements according to a given scenario.
8.2 Variables	8.2.1 Handling Variables	5) Identify between different types of variables for a given scenario. *Limited to: integer, float, String and Boolean.* 6) Interpret a program snippet that includes the use of variables and/or input statements. *Program-snippets may include variable initialization.* *Limited to:* • *input() function;* • *variables as listed in 8.2.1 (5).* 7) Interpret a program snippet that includes the use of variables and/or type conversion functions and/or input statements. *Limited to int(), float(), and str() type conversions.*

Topic	Sub-Topic	Assessment Criteria
		8) Develop a program that includes the use of variables and/or input statements. *Limited to those listed in (6)*. 9) Develop a program using variables and/or type conversion functions and/or input statements. *Limited to those listed in (7)*.
8.3 Arithmetic Operations		10)Identify arithmetic operations in a given scenario. *Limited to the following arithmetic operators:* • *= equals, / division, + addition, // floor division,* • *- subtraction, % modulus, * multiplication, ** power of.* 11)Interpret a program snippet that includes arithmetic operations. 12)Interpret a program snippet that includes arithmetic assignment operators. *Limited to +=, -=, *=, /=, //=, %=.* 13)Develop a program using arithmetic operations. *Limited to those listed in (10).* 14)Develop a program using arithmetic assignment operators. *Limited to those listed in (12).*
8.4 Decision	8.4.1 Decision Statements	15)Interpret a program snippet that includes decision statements. *Limited to:* • *Decision statements: if, elif, else* • *Conditional operators: ==, !=, >, <, >=, <=* • *Logical Operators: and, or, not* • *Membership operators: in, not in*

Topic	Sub-Topic	Assessment Criteria
		16)Interpret a program snippet that includes nested decision statements. 17)Develop a program using decision statements. *Limited to those listed on (15).* 18)Develop a program using nested decision statements.
8.5 Iteration	8.5.1 Iteration statements	19)Interpret a program snippet that includes iteration statements. *Limited to for-in loop, while loop, and while-else loop.* *Note: the for-in loop may include the range() function.* 20)Interpret a program snippet that includes nested iteration statements. *Note: may include 'break' and 'continue' statements.* 21) Develop a program using iteration statements. *Limited to those in (19).* 22)Develop a program using nested iteration statements.
8.6 Functions	8.6.1 Using and defining functions	23)Define the term function and/or module. 24)Distinguish between function and module. 25)Distinguish between built-in function and user defined function. 26)Interpret a program snippet that includes String functions. *Limited to: Get character from string (String[position]), Substring (String[2:5]), strip(), len(), lower(), upper(), replace(), find(), count().* 27)Interpret a program snippet that includes user defined functions. 28)Develop a program using String functions. 29)Develop a program using user-defined functions. *Note: user-defined functions may include value return and parameters.*

Topic	Sub-Topic	Assessment Criteria
8.7 Data Structures	8.7.1 Lists, Tuples, and Dictionaries	30)Interpret a program snippet that includes tuples and/or lists as data structures. *Limited to the use of the following functions for tuples: max(), min(), count(), index(), sum(), sorted(), and len()* *Limited to the use of the following functions for lists: append(), clear(), count(), insert(), len(), pop(), remove(), reverse(), and sort().* 31)Interpret a program snippet that includes dictionaries as data structure. *Limited to the use of the following functions: clear(), items(), get(), and pop().* 32)Develop a program using tuples and/or lists as data structures. *Limited to those in (30).* 33)Develop a program using dictionaries as data structures. *Limited to those in (31).*
8.8 Modules	8.8.1 Random module	34)Interpret the statement import and/or fromimport and/or from-import-as. *Examples: ▪ import random ▪ from random import randint ▪ from random import * ▪ from random import randrange as num_generator* 35)Interpret a program snippet that includes the Random module. *Limited to randint(start, stop) function.* 36)Interpret a program snippet that includes the Random module to access and modify items in data structures. *Limited to: choice(), shuffle() and sample() from a List or Tuple only.* 37)Develop a program using the Random module. *Limited to those in (35)*.

Topic	Sub-Topic	Assessment Criteria
		38)Develop a program using the Random module to access and/or modify items in data structures. *Limited to those in (36).*
8.9 Using text files		39)Develop a program to save text (Strings) to a text file. *Limited to: open() with "r", "a", and "x" modes, read(), close().* 40)Develop a program to write a list or a dictionary to a file.
8.10 Programming paradigms		41)Define a paradigm. 42)List the following programming paradigms: ● imperative; ● declarative; ● object oriented; ● multi-paradigm; ● low-level paradigm. 43)Compare and contrast the paradigms listed in (42).
8.11 Object Oriented Programming (OOP)	8.11.1 Classes and Instances	44)Outline the structure of a class. *In terms of attributes and methods.* 45)Outline the concept of constructor methods. 46)Define the 'self' keyword. 47)Relate classes and objects/instances. 48)Interpret a program using 44) to 47). 49)Develop a program using 44) to 47).

Topic	Sub-Topic	Assessment Criteria
	8.11.2 Encapsulation	50)Outline the concept of encapsulation. 51)Interpret a program using private attributes and public get( ) and set( ) methods. 52)Develop a program using private attributes and public *get( )* and *set( )* methods.
	8.11.3 Inheritance	53)Outline the concept of inheritance. *In terms of super/parent classes and sub/child classes.* 54)Outline the concept of method overriding. 55)Define the 'super' keyword. 56)Interpret a program using 53) to 55). 57)Develop a program using 53) to 55).
8.12 Programming Errors and Exceptions	8.12.1 Types of programming errors	58)Distinguish between syntax, logical and runtime errors. 59)Explain the effect of an error on program compilation and/or runtime. *Limited to:* ● *syntax: does not compile;* ● *logical: gives undesired result;* ● *runtime: stops executing.*
	8.12.2 Handling exceptions	60)Explain the use of a 'try..except..finally' block. 61)Interpret a program using exception handling. *Limited to ZeroDivisionError, FileNotFoundError, ValueError.* 62)Develop a program using (60).
	8.12.3 Debugging	63)Outline commonly used debugging tools.

Topic	Sub-Topic	Assessment Criteria
		Limited to logs, breakpoints and stepping, and variable inspection.

Subject Focus:	**Databases**
Learning Outcome 9:	**I can apply foundational database concepts, including structures, DBMS role, three-level architecture, relational model, E-R modelling, normalisation, and basic SQL.**

Topic	Sub-Topic	Assessment Criteria
9.1 Introduction to Databases and Database Management Systems (DBMS)	9.1.1 Introduction to databases	1) Define a database. 2) Define a flat file database. 3) Identify limitations of a flat file database.
	9.1.2 Introduction to DBMS	4) Define a DBMS. 5) Describe the structure of a DBMS. *To include schema, data dictionary, and database languages (Data manipulation language (DML) and Data description language (DDL).* 6) Differentiate between DML and DDL. 7) Describe the functions of a DBMS. *Limited to data storage retrieval and update, backup and recovery services, integrity services, provides an interface, and security services.*
9.2 Database Administrator (DBA)		8) Define the role of the DBA. 9) List the responsibilities of the DBA.
9.3 The three-level architecture		10)Justify the need for the three-level architecture. 11)Outline the three-level architecture of a DBMS: ● External (user view); ● Conceptual (logical view); ● Internal (physical view). 12)Recommend possible user groups (in the external view) for a given scenario.

Topic	Sub-Topic	Assessment Criteria
9.4 Relational Databases		13)Define a relational database. 14)Define the basic terms: ● record/tuple; ● field/attribute; ● primary key; ● secondary key; ● foreign key; ● composite key; ● entity. 15)Describe the logical structure of a relational database as a set of tables linked together using common fields. 16)Describe the different types of cardinality. *Limited to one-to-one, one-to-many, many-to-many.* 17)Write schemas using the short/shorthand notation.
9.5 Entity-Relationship (E-R) Diagrams The standard Crow's Foot notation is to be used (Appendices)		18) Interpret an E-R Diagram. *Limited to five entities.* 19)Draw E-R Diagram from a given scenario. *Limited to five entities.*
9.6 Normalisation		20)Define Normalisation. 21)Explain the need for normalisation. 22)Describe the 1st, 2nd and 3rd normal forms. 23)Normalise a simple flat file database, up to the third normal form (3NF).

Topic	Sub-Topic	Assessment Criteria
9.7 Structured Query Language (SQL)		24)Define Query language. 25)Interpret SQL commands. *Limited to:* *SELECT… FROM...;* *SELECT… DISTINCT...FROM...;* *SELECT… FROM…WHERE;* *SELECT… FROM…WHERE…AND/OR/NOT;* *SELECT… FROM…WHERE…ORDER BY… (using ascending or descending order);* *SELECT… FROM…WHERE… GROUP BY… (using COUNT, SUM, MAX, MIN, AVG);* *SELECT… FROM… INNER JOIN…. LEFT/RIGHT/FULL OUTER JOIN…;* *SELECT… FROM… WHERE… GROUP BY… HAVING… ORDER BY…*

Scheme of Assessment

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

1. Written examination papers:
   - Paper I;
   - Paper II.
2. Portfolio.

Component	Paper	Section	Learning Outcome	% Weighting
Written Examination Papers	Paper I	All Sections	All LOs	40%
	Paper II	Section A	LO 8	15%
		Section B	All LOs excluding LO 8	30%
Portfolio		Refer to notes hereunder	LO 8	15%

Written Examination Papers

- candidates will write their answers on a separate booklet provided by MATSEC;

Paper I

- carries a total of 100 marks amounting to 40 % of the total score;
- consists of twenty short compulsory questions. These questions will require short and to the point answers each worth 5 marks.

Paper II

- carries a total of 90 marks amounting to 45 % of the total score;
- consists of two sections:

Section A

- carries a total of 30 marks amounting to 15% of the total score;
- consists of one compulsory question focused on LO 8.

Section B

- carries a total of 60 marks amounting to 30% of the total score;
- consists of six long questions of which candidates are expected to answer three. Each question carries 20 marks.

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.

General Guidelines (5%)

- Candidates are expected to provide **ONE** program for each section 1, 2, 3 and 5, and **THREE** programs for section 4, as explained hereunder.
- The portfolio is to be submitted according to directions set by MATSEC.
- 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 (5%)

Input/Output:

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

Variables:

- 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 (10%)

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: (5%)

- 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 (20%)

Iteration: (10%)

- 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.
- Use of at least **ONE** nested iteration statement.

Functions: (10%)

- 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, Modules, and File Handling (30%)

- Candidates are to present a separate program for each data structure as listed below.
- (20%) One program should also demonstrate the random module and file handling.
- (5%, 5%) Two remaining programs.

Data Structures:

- Use of a list including at least the implementation of **THREE** functions (e.g., append(), clear(), count(), insert(), len(), pop(), remove(), reverse(), and sort()).
- Use of a tuple including at least the implementation of THREE functions (e.g., max(), min(), count(), index(), sum(), sorted(), and len())
- Use of a dictionary including at least the implementation of **THREE** functions (e.g clear(), items(), get(), and pop()).

Modules:

- Use of the random module and its associated functions.

File Handling:

- Saving and retrieving data to/from text files

Section 5: Object-Oriented Programming and Error Handling (30%)

Object-Oriented Programming (OOP): (25%)

- Use of at least **THREE** classes, including attributes, methods, and constructors.
- Use of encapsulation, inheritance, and method overriding.

Programming Errors and Exceptions: (5%)

- Use of at least **ONE** try, except, and finally blocks for exception handling.

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

Data Transfer instructions	MOV	Moves byte or word to register or memory
	PUSH	Push a word on stack
	POP	Pop a word from stack
Logical Instructions	NOT	Logical not (1's complement)
	AND	Logical and
	OR	Logical or
	XOR	Logical exclusive-or
Arithmetic Instruction	ADD , ADC	Add and Add with carry
	SUB, SBB	Subtract and Subtract with borrow
	INC	Increment
	DEC	Decrement
	CMP	Compare
Transfer Instructions	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
Flag Manipulation	CLC	Clear Carry
	STC	Set Carry
	SHL, SHR	Logical Shift Left or Right
	RCL, RCR	Rotate through Carry Left or Right
Shift and Rotate	HALT, END	Pseudo-directives

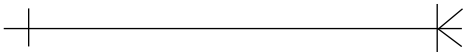
List of Acronyms

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

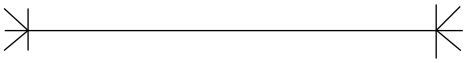
Crow's Foot E-R Diagram Notation

1 – 1

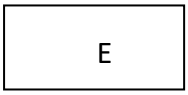
1 – M



M – N



Data Entity



"E" is entity's name

Boolean Algebra

1. Commutative Laws
   a) $A + B = A + B$
   b) $A \,.\, B = B \,.\, A$
2. Associative Laws
   a) $A + (B + C) = (A + B) + C$
   b) $A \,.\, (B \,.\, C) = (A \,.\, B) \,.\, C$
3. Distributive Laws
   a) $A \,.\, (B + C) = A \,.\, B + A \,.\, C$
   b) $A + B \,.\, C = (A + B) \,.\, (A + C)$
4. De Morgans Laws
   a) $\overline{(A + B)} = \overline{A} \,.\, \overline{B}$
   b) $\overline{(A \,.\, B)} = \overline{A} + \overline{B}$
5. Laws of Absorption
   a) $A + A \,.\, B = A$
   b) $A \,.\, (A + B) = A$
6. Double Complement Law
   a) $\overline{\overline{A}} = A$
7. Laws of Tautology
   a) $A + A = A$
   b) $A \,.\, A = A$
   c) $A + \overline{A} = 1$
   d) $A \,.\, \overline{A} = 0$
   e) $A + 1 = 1$
   f) $A \,.\, 1 = A$
   g) $A + 0 = A$
   h) $A \,.\, 0 = 0$