



**L-Università
ta' Malta**

**MATSEC
Examinations Board**



Examiners' Report

SEC Computing

Main Session 2025

TABLE OF CONTENTS

A.	STATISTICAL INFORMATION	1
B.	GENERAL REMARKS	1
	Controlled Paper I (L 1-2-3)	1
	Controlled Paper II.....	1
C.	COMMENTS ON CANDIDATES' PERFORMANCE	2
	Controlled Paper I (L 1-2-3)	2
	Controlled Paper II (L 1-2).....	3
	Controlled Paper II (L 2-3).....	4

A. STATISTICAL INFORMATION

A total of 698 candidates registered for SEC09 Computing, a decrease of 43 candidates compared to 2024. Of these, 674 candidates chose Level 2–3, and 24 candidates chose Level 1–2. There were 645 school candidates, and 53 private candidates.

Table 1 shows the distribution of levels of the Main 2025 Session of the examination.

Level	Number of Candidates
1	2
2	71
3	399
na	226
Total	698

Table 1: Distribution of Levels

B. GENERAL REMARKS

Controlled Paper I (L 1-2-3)

This paper assessed candidates across two areas of computing, including Arduino-based systems programming and Python software development.

Section A was a written task focusing on an Arduino scenario. It evaluated candidates' ability to recall programming constructs, analyse code segments, apply modifications, and evaluate system behaviour through tasks. It included fill-in-the-blank code completion, explanation of code purpose, flowcharting, and reasoning-based questions. Candidates generally found this section moderately challenging, with approximately half performing well and demonstrating clear understanding, while the remainder found certain analytical and reasoning tasks more difficult.

Section B was an applied programming task using Thonny IDE, where candidates were expected to write a modular Python program from scratch, assessing application, synthesis, and evaluation skills through coding tasks involving input validation, code logic, user-defined functions, data structure implementation, and adherence to good programming practices such as meaningful variable names, proper indentation, and a user-friendly interface. This section proved more demanding, with a wider spread in performance. In fact, stronger candidates were able to produce well-structured, functional solutions, while others faced difficulty implementing the required logic and coding practices.

Controlled Paper II

This exam assessed candidates across computing concepts, embedded systems, digital systems, and programming concepts. All questions were written tasks.

In papers targeting Level 1-2 and Level 2-3 candidates, the scenarios were identical in topic. However, in the Level 2-3 paper, scenarios were broader and required extended reasoning, integration of

multiple syllabus areas, and evaluation of alternative solutions. In contrast, the Level 1-2 paper featured shorter, more segmented scenarios that primarily supported factual recall or straightforward application, with sub-questions providing scaffolding to guide responses.

Overall, Section A evaluated understanding of networked systems, IoT devices, operating systems, digital logic, assembly programming, and firmware, requiring candidates to recall facts, analyse scenarios, apply concepts, and evaluate solutions through short-answer questions, comparisons, justifications, calculations, binary conversions, and reasoning-based questions.

Section B assessed applied knowledge of databases, Python programming concepts, embedded systems, microcontrollers, and networking system reasoning and evaluation. Candidates were required to analyse code snippets, suggest data validations, interpret algorithms, represent logic using flowcharts and logic gates, and recommend system or hardware solutions. Tasks emphasized application, synthesis, and evaluation skills while reinforcing good programming and system analysis practices, such as modular thinking, meaningful naming, proper indentation, and readability.

This was the first examination paper set according to the new syllabus. Overall, the paper provided a fair assessment for the Level 2–3 cohort while it proved more challenging for Level 1–2 candidates. It maintained a comparable level of difficulty to previous examinations but adopted a different format, combining questions of varying complexity to differentiate candidate performance. The paper generally supported candidates' recall and comprehension skills, particularly in theoretical questions. However, scenario-based and applied tasks, especially those involving programming, problem-solving, and analytical reasoning, proved more demanding.

C. COMMENTS ON CANDIDATES' PERFORMANCE

Controlled Paper I (L 1-2-3)

Question 1

Overall, Question 1 was manageable for most candidates, though performance varied considerably. Some candidates showed good knowledge of Arduino syntax and program flow, completing the missing code and explaining constructs with confidence. Other candidates, however, often confused Arduino with other programming languages, particularly in variable initialisation and Boolean logic.

While most managed the flowchart and good programming practices, difficulties arose in explaining the servo's initial position, the role of tracing statements, and the use of Boolean flags to prevent repeated switching. The final part, which introduced the potentiometer component, showed that many candidates did not properly grasp understanding of hardware concepts such as ADC resolution and value ranges.

Question 2

Overall, responses indicate that Question 2 was more demanding, with few candidates managing to produce a complete solution. Most coped well with the setup section and showed awareness of basic coding practices, such as clear variable names and proper indentation. However, many struggled in the main game logic, where mistakes in program flow and incomplete solutions were common.

Errors often stemmed from confusion over data structures, with some using lists or sets instead of tuples for valid colours, and from difficulties in generating the secret code correctly. While input handling was usually attempted, issues with spaces, validation, and case sensitivity were frequent and weakened later parts of the game.

The greatest challenge lay in comparing guesses with the secret code and applying the scoring system. Although some marks were earned for deducting points and counting attempts, many could not implement the full logic for checking guesses accurately. As a result, only a small number of candidates managed to deliver a working game.

Controlled Paper II (L 1-2)

The following table shows the distribution of grades for Paper 1-2.

GRADE	1	2	3	4	5	6	7	8	U	ABS	TOTAL
PAPER 1-2	-	-	-	-	-	2	-	2	8	12	24

Question 1

Overall, this question proved challenging for most candidates. The majority were unable to clearly identify two advantages of networked devices, and very few distinguished correctly between embedded operating systems and general-purpose systems. Only a single candidate was able to define the three types of user interfaces accurately. Most candidates also showed limited understanding of backup and restore functionalities, and while some could identify one advantage of using a database, very few provided both.

Question 2

Most candidates struggled to describe the concept of the Internet of Things (IoT). Very few correctly identified advantages and disadvantages of Ethernet, and many were unaware that a network card enabled device connectivity. While some could recommend network devices for Internet access or Wi-Fi extension, only one candidate showed understanding of the difference between MAC and IP addresses. Knowledge of IPv4 versus IPv6 was also largely lacking.

Question 3

Only a few candidates correctly identified high- and low-level programming languages. Many did not know the definitions of source and executable code, and few could list the language translators. Most candidates correctly classified the game interface and firmware as system or application software, but there was a common lack of understanding of software licenses.

Question 4

Most candidates could not explain why digital systems use binary. Only some completed decimal-to-binary conversions correctly, and many did not perform bit-shift operations for multiplication or division. Binary addition and subtraction were also frequently answered incorrectly.

Question 5

Candidates struggled to identify table names and foreign keys, but primary keys were generally correct. Most candidates provided incorrect SQL statements, with common errors in query structure. Python programming questions indicated poor understanding of data structures, loops, string handling, and data validation.

Question 6

Many candidates could not identify secondary storage capacity, NAND/NOR gate symbols, or filing systems. Understanding of display resolution, bit depth, CPU word length, and address bus was also limited. Assembly language questions were often answered incorrectly, with partial understanding of opcodes, operands, and the roles of the CU, ALU, and accumulator.

Question 7

Most candidates performed poorly, showing limited understanding of algorithms, flowcharts, and logic circuits. Few were able to define microcontrollers, analogue and digital data, or ADCs. Only a small number correctly completed the flowchart or designed the logic circuit to control the thermal camera.

Controlled Paper II (L 2-3)

The following table shows the distribution of grades for Paper 2-3.

GRADE	1	2	3	4	5	6	7	8	U	ABS	TOTAL
PAPER 2-3	39	80	123	72	50	44	63	-	142	61	674

Question 1

Most candidates correctly identified two advantages of networked devices and distinguished between embedded and general-purpose operating systems. A large portion accurately described at least two types of user interfaces, though some had difficulty differentiating CLI from GUI and NUI. Many demonstrated an understanding of the backup and restore utility, providing suitable use cases. In contrast, only a portion of candidates provided both advantages of using a database over manual record-keeping, with some giving only one. Despite minor challenges, the overall performance was positive.

Question 2

Most candidates demonstrated a basic understanding of the Internet of Things (IoT), though many did not explicitly mention the role of the Internet. The majority correctly identified an advantage and a disadvantage of using an Ethernet connection, and many accurately distinguished between MAC and IP addresses, as well as IPv4 and IPv6. Candidates performed well in recommending a network

device to connect the server to the internet. However, many struggled with identifying the correct network component within each IoT device and with suggesting solutions to enhance Wi-Fi connectivity across floors, often providing incorrect answers. Overall, just over half of the candidates achieved an average mark, with a notable number performing poorly in certain parts of the question.

Question 3

Candidates struggled with distinguishing between high-level and low-level programming languages. Many could not explain why a 4th Generation Language is unsuitable for developing a game interface, though most could justify why assembly language is more appropriate for firmware. The concept of translating source code into executable code was partially understood, but not the same with the identification of the correct translator (assembler vs. compiler) for the firmware and game interface. Most candidates correctly classified the game console interface as application software, and many provided valid reasons for using cloud-based gaming services, although fewer could recommend the appropriate software license.

Question 4

Candidates performed well in explaining why binary is used, converting decimal to binary, and performing simple bit-shift operations. Difficulties arose in applying two's complement for binary subtraction, and many could not explain why a Half Adder is unsuitable for 8-bit addition. Basic binary operations were mostly handled correctly, but advanced logic concepts and 8-bit arithmetic proved challenging.

Question 5

Candidates were generally able to explain the need for a database and, in some cases, identify foreign keys and justify table relationships. However, the Python data validation section was poorly handled, particularly identifying the correct data structure, distinguishing between slices and single elements, and modifying code to accept both uppercase and lowercase letters. SQL-related questions were also difficult, with many candidates unable to write correct queries or include ordering. Most candidates understood the need for a firewall and HTTPS connection, though some could only justify one security measure. This question proved to be very challenging overall.

Question 6

Candidates showed a basic understanding of ROM versus RAM usage and the overall function of the assembly program. However, questions regarding universal gates, instruction sets, CPU components, buses, and accumulator functions were poorly answered. Many confused bus and register names or did not identify control signals correctly. Writing Python equivalents of assembly variables was generally handled well. In general, this question highlighted gaps in knowledge of microcontroller architecture and assembly language.

Question 7

Most candidates justified the use of a microcontroller and distinguished analogue from digital data. Parallel data transfer and ADC-related reasoning were poorly answered. Many struggled with Python-related tasks such as identifying infinite loops, fixing syntax errors, and drawing accurate flowcharts.

Examiners' Report (201925): SEC Computing

Questions about logic gates, including truth tables, circuit diagrams, and Boolean expressions, were also challenging. Overall, candidates found the integration of microcontroller hardware, Python programming, and digital logic difficult.

Chairperson

Examination Panel 2025