Coimisiún na Scrúduithe Stáit
State Examinations Commission

Leaving Certificate 2026

Marking Scheme

Computer Science

Higher Level

Note to teachers and students on the use of published marking schemes

Marking schemes published by the State Examinations Commission are not intended to be standalone documents. They are an essential resource for examiners who receive training in the correct interpretation and application of the scheme. This training involves, among other things, marking samples of student work and discussing the marks awarded, so as to clarify the correct application of the scheme. The work of examiners is subsequently monitored by Advising Examiners to ensure consistent and accurate application of the marking scheme. This process is overseen by the Chief Examiner, usually assisted by a Chief Advising Examiner. The Chief Examiner is the final authority regarding whether or not the marking scheme has been correctly applied to any piece of candidate work.

Marking schemes are working documents. While a draft marking scheme is prepared in advance of the examination, the scheme is not finalised until examiners have applied it to candidates' work and the feedback from all examiners has been collated and considered in light of the full range of responses of candidates, the overall level of difficulty of the examination and the need to maintain consistency in standards from year to year. This published document contains the finalised scheme, as it was applied to all candidates' work.

In the case of marking schemes that include model solutions or answers, it should be noted that these are not intended to be exhaustive. Variations and alternatives may also be acceptable. Examiners must consider all answers on their merits, and will have consulted with their Advising Examiners when in doubt.

Future Marking Schemes

Assumptions about future marking schemes on the basis of past schemes should be avoided. While the underlying assessment principles remain the same, the details of the marking of a particular type of question may change in the context of the contribution of that question to the overall examination in a given year. The Chief Examiner in any given year has the responsibility to determine how best to ensure the fair and accurate assessment of candidates' work and to ensure consistency in the standard of the assessment from year to year. Accordingly, aspects of the structure, detail and application of the marking scheme for a particular examination are subject to change from one year to the next without notice.

Marking Scheme - Section C

Structure of the marking scheme for Section C (Programming)

Candidate responses are marked according to different scales, depending on the types of response anticipated. For example, scales labelled B divide candidate responses into three categories (correct response, response with some merit, and response of no substantial merit), and so on. The scales and the marks that they generate are summarised in this table:

Scale Label	A	B	C	D
No. of categories	2	3	4	5
5 mark scale	0, 5		0, 2, 3, 5	
10 mark scale		0, 7, 10	0, 3, 7, 10	0, 3, 5, 7, 10

A general descriptor of each point on each scale is given below. More specific directions in relation to interpreting the scales in the context of each question are given in the scheme, where necessary.

Marking scales - level descriptors

A-scales (2 categories)

- response of no substantial merit
- correct response

B-scales (3 categories)

- response of no substantial merit
- response with some merit
- correct response

C-scales (4 categories)

- response of no substantial merit
- response with some merit
- almost correct response
- correct response

D-scales (5 categories)

- response of no substantial merit
- response with some merit
- response about half-right
- almost correct response
- correct response

Section A — Short Answer Questions — 54 marks

Answer any **nine** questions.

Question 1 **6 marks**

a	b
0	7
1	**6**
3	**4**
5	**2**
7	0

Each correct response 1 mark

Question 2 **6(3,3) marks**

Purpose

Both cache and RAM are volatile memory used to temporarily store data and instructions needed by the CPU. However, cache stores frequently accessed data close to the CPU to reduce access times and improve performance, whereas RAM stores the larger set of programs and data currently being used by the operating system and applications.

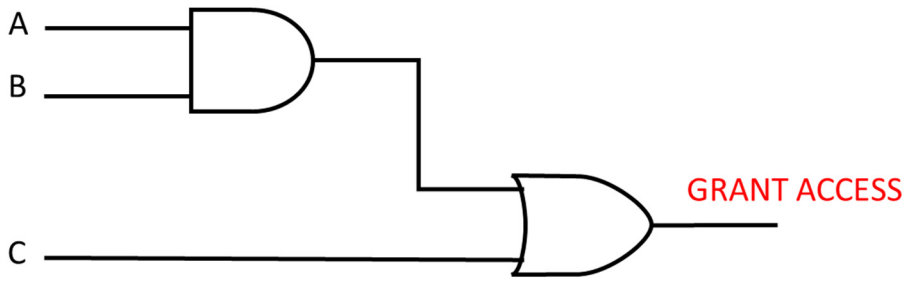
Speed

Both cache and RAM provide fast access to data, but cache is significantly faster than RAM. This allows the CPU to retrieve frequently used data more quickly from cache, while RAM provides slower access to a larger amount of working memory.

For each feature:

Very good explanation - clear understanding demonstrated 3 marks
Fair explanation - limited understanding 2 mark

Question 3 **6 marks**



Correct AND gate	1 mark
Correct OR gate	1 mark
A connected to AND	1 mark
B connected to AND	1 mark
Output of AND connected to OR	1 mark
C connected to OR	1 mark

Question 4 **6(2,4) marks**

(a)

FFFF

Correct answer	2 marks
Response with some merit	1 mark

(b)

0011 1111

Correct response	4 marks
Correct answer without leading zeros	3 marks
Response with some merit	2 mark

Question 5 **6(3,3) marks**

(a)

- **Data can be accessed from anywhere with an internet connection**
- **The school does not need to maintain its own physical servers**
- **Cloud storage providers usually handle backups and updates automatically. This reduces the risk of data loss and lowers maintenance effort for the school**
- **Any reasonable response e.g. security, scalability, elasticity**

(b)

- **Student data security and privacy could be at risk if the cloud is hacked**
- **The school depends on a reliable internet connection to access records**
- **There may be legal or data protection rules the provider must follow**
- **Any reasonable response e.g. cost**

For each part:

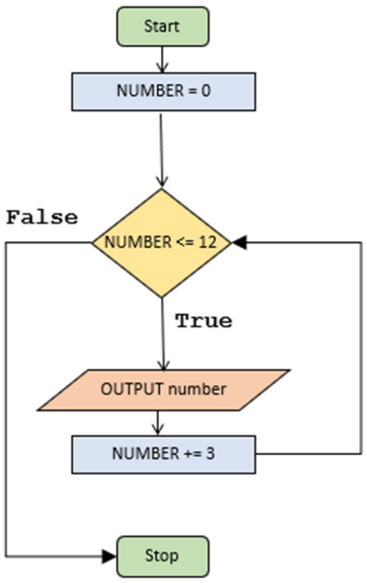
Very good explanation - clear understanding demonstrated	3 marks
Fair explanation - limited understanding	2 mark

Question 6 **6 marks**

Pseudocode:

Initialise number to 0
Loop while number ≤ 12
Display number
Add 3 to number

Flowchart



Full correct response	6 marks
Response with some merit	4 marks
Response with limited merit	2 marks

Question 7 **6(4,2) marks**

(a)

Pass 5:	**4**	**5**	**6**	**9**	**12**	10	3	11
Pass 6:	**4**	**5**	**6**	**9**	**10**	**12**	3	11
Pass 7:	**3**	**4**	**5**	**6**	**9**	**10**	**12**	11
Pass 8:	**3**	**4**	**5**	**6**	**9**	**10**	**11**	**12**

Each correct pass: 1 mark

(b)

- **Insertion sort works by taking one item at a time and inserting it into the correct position in a list that is already partly sorted. The item is compared with the values before it and moved (swapped) left until it is in the correct place. This process continues until all items in the list are in order.**

Very good response - clear understanding demonstrated 2 marks
Fair response - limited understanding 1 mark

Question 8 **6(3,3) marks**

- **AI does not truly understand topics; it predicts answers based on patterns, which can sometimes lead to incorrect or misleading responses (i.e. problem is inherent in the way AI works)**
- **AI is trained on large amounts of data, which can include errors or outdated information, so it may repeat these mistakes (i.e. problem relates to training data)**
- **AI can be biased because it learns from data created by humans, which may contain unfair or one-sided views. This can cause AI to give unbalanced or misleading information (i.e. problem relates to training data)**

For each explanation:
Very good explanation - clear understanding demonstrated 3 marks
Fair explanation - limited understanding 2 mark

Question 9 **6(3,3) marks**

(a)

- **Modular design in software development means dividing a large program into smaller, manageable sections called modules. Each module is responsible for a specific task, such as handling orders or payments. This makes the system easier to develop, test, and maintain.**

Very good explanation - clear understanding demonstrated	3 marks
Fair explanation - limited understanding	2 mark

(b)

- **Order taking module to record customer orders**
- **Payment processing module to handle bills and payments**
- **Menu management module to update food items and prices**
- **User account module to manage staff logins and roles**
- **Reporting module to track sales and generate reports**
- **Table management module to track table availability**
- **Inventory module to monitor ingredient stock levels**
- **Any reasonable response**

For each module correctly identified	1 mark

Question 10 **6(3,3) marks**

- **Equitable Use: All students access the same learning content regardless of ability, for example a visually impaired student uses a screen reader while receiving the same information as other students.**
- **Flexibility in Use: Students can choose whether instructions are presented as text, audio, or video.**
- **Simple and Intuitive Use: Navigation uses clear, consistent menus so students can use the app without training.**
- **Perceptible Information: Important alerts are provided using both visual and audio cues/ Important instructions are provided as both text and audio, allowing students with visual or auditory impairments to access the information.**
- **Tolerance for Error: Students can undo an action or confirm before deleting work.**
- **Low Physical Effort: Large buttons reduce the need for precise mouse or touchscreen movements.**
- **Size and Space for Approach and Use: The interface supports zooming and sufficient spacing between controls, so they are easy to select.**

For each correctly named principle	2 marks
For each description:	
Good description - clear understanding demonstrated	1 mark

Question 11 **6(4,2) marks**

(a)

- **Maintain a minimum following distance based on current speed.**
- **Reduce speed when visibility decreases.**
- **Increase stopping distance in wet or icy conditions.**
- **Choose the route with the least predicted congestion.**
- **Leave extra space when driving near cyclists or pedestrians.**
- **Any reasonable response - a good solution found in a short amount of time**

Very good response - clear understanding demonstrated	4 marks
Fair response - limited understanding	2 marks

(b)

- **Heuristics do not always give the best or correct decision**
- **Decisions are based on simple rules, not full understanding**
- **Heuristics can be biased by the data or rules they are based on**
- **They may oversimplify complex real-world situations**
- **Different heuristics can sometimes conflict with each other**
- **Any reasonable response**

Correctly response	2 marks

Question 12 **6 (3,3) marks**

(a)

- **The controller follows the Turing machine's rules. Based on the current state and the symbol being read, it decides what symbol to write, whether to move the read/write head left or right, and what the next state should be, including when to halt.**

(b)

- **Turing machines are important because they give us a simple way to describe how any computer can work. They help computer scientists understand what kinds of problems can be solved by computers and what problems cannot be solved at all. Even though they are very simple, Turing machines are powerful enough to represent any modern computer, which is why they are so central to Computer Science.**

For each part:

Very good response - clear understanding demonstrated	3 marks
Fair response - limited understanding	2 marks

Section B — Long Questions — 76 marks

Answer any **two** questions.

Question 13 — **38 (13, 12, 13) marks**

(a) — **13 (3, 2, 8) marks**

(i) — **3 marks**

- **An information system (IS) is the software used to help organise and analyse business data and streamline the day to day (transactional) operation of a business.**
- **An information system is a system that takes in data, processes it, and produces useful information.**

Very good explanation - clear understanding demonstrated 3 marks
Fair explanation - limited understanding 2 marks

(ii) — **2 (1, 1) marks**

Any two from the following:

- **Hardware - physical devices like computers and servers**
- **Software - programs and applications used by the system**
- **Data - raw facts that are stored, processed and output as information**
- **People - users who interact with the system**
- **Processes (procedures) - rules and steps for using the system**

For each correct response 1 mark

(iii) — **8 (3, 1, 3, 1) marks**

<u>Functional Requirements</u>

- **Allow users to submit an online application for the lottery**
- **Send confirmation or rejection notifications to applicants**
- **Store and manage applicant details securely**
- **Randomly select winners and generate a list of successful applicants**
- **Allow administrators to view and manage, applicants and results**
- **Data validation e.g. prevent duplicate applications from the same person**
- **Any reasonable requirement**

<u>Non-Functional Requirements</u>

- **The system should be easy to use and accessible to all users**
- **The system should respond quickly, even with many users**

- **The system should be secure to protect personal data**
- **The system should be reliable and available during high demand**
- **The system should comply with data protection laws**
- **The system should be scalable to handle increased traffic**
- **Any reasonable requirement**

For best requirement (functional and non-functional):
Good response –understanding demonstrated 3 marks

For second requirement (functional and non-functional):
Good response - understanding demonstrated 1 mark

(b) **12 (6, 6) marks**

(i) **6 (2, 2, 2) marks**

String

- **Name: Stored as a string because it contains letters and may include spaces**
- **Email address: Stored as a string because it uses letters, symbols, and numbers**
- **Home address: Stored as a string because it is text made up of words and spaces**
- **Any reasonable example**

Integer

- **Age: Stored as an integer because it is a whole number**
- **Number of tickets requested: Stored as an integer because it counts items**
- **Year of birth: Stored as an integer because it uses whole numbers only**
- **Any reasonable example**

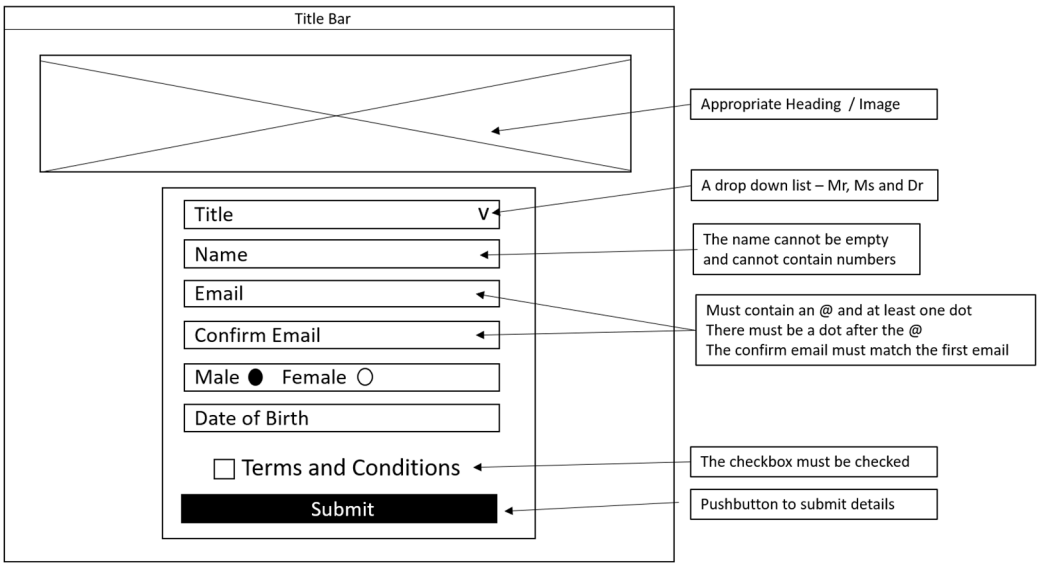
Boolean

- **Agree to terms and conditions: Boolean because the answer is either true or false**
- **Resident of Ireland: Boolean because the response is yes or no**
- **Over 18 years old: Boolean because there are only two possible values**
- **Any reasonable example**

For each datatype:
Correct example 1 mark
Good explanation - understanding demonstrated 1 mark

(ii) **6 marks**

Sample Wireframe



Clear heading/image	1 mark
Controls / Widgets:	
Very good examples - clear understanding demonstrated	2 marks
Fair example - limited understanding	1 mark
Annotations / Notes:	
Very good response - clear understanding demonstrated	2 marks
Fair response - limited understanding	1 mark
Submit pushbutton	1 mark

(c) **13 (4, 9) marks**

(i) **4 (2, 2) marks**

User Experience

- **A mobile application can provide a smoother and faster experience because it is designed specifically for mobile devices and can use features such as notifications and touch controls.**
- **A responsive website works through a web browser and adjusts to different screen sizes. It may not feel as interactive as an app, but users can access it instantly without installing software.**
- **Any valid response**

Accessibility

- **A responsive website is more accessible because it can be used on phones, tablets, laptops, and desktop computers using any web browser.**
- **A mobile application may require a compatible device and storage space, but it can include accessibility features such as voice input, larger text, and screen reader support.**
- **Any valid response**

For each heading:

Very good explanation - clear understanding demonstrated	2 marks
Fair explanation - limited understanding	1 mark

(ii) **9 (5, 2, 2) marks**

Validation examples (x 2)

- **Range check: Ensure age entered is within a valid range, for example 18-120**
- **Format check: Make sure the email address includes an @ symbol**
- **Presence check: Ensure required fields, such as name or agreement to terms, are not left blank**
- **Length check: Ensure a phone number has the correct number of digits**
- **Type check: Make sure numeric fields, such as year of birth, only contain numbers**
- **Any reasonable example**

Verification example

- **Double entry: Applicants enter their email address twice to make sure both entries match**
- **Visual check: Applicants review their details on a confirmation screen before submitting**
- **Email confirmation: A confirmation email is sent so applicants can check their details are correct**
- **Any reasonable example**

For best check (validation or verification):

Good explanation - understanding demonstrated	5 marks

For other checks (validation or verification):

Good explanation – clear understanding demonstrated	2 marks

Question 14 **38 (16, 12, 10) marks**

(a) **16 (3, 6, 4, 3) marks**

(i) **3 marks**

- **The path is not continuous - after crossing bridge 6 you are on land area B, but bridge 7 does not connect to B, so the next crossing is not possible.**

Very good explanation - clear understanding demonstrated 3 marks
Fair explanation - limited understanding 2 marks

(ii) **6 (3, 3) marks**

- **Euler Paths from Figure 4 (start at land area C):**
  - **1→ 3 → 4 → 2**
  - **2→ 4 → 3 → 1**
- **Euler Paths from Figure 5 (start at land area C):**
  - **1→ 3 → 4 → 2 → 5**
  - **1→ 3 → 5 → 2 → 4**
  - **2→ 4 → 5 → 1 → 3**
  - **2→ 4 → 3 → 1 → 5**
  - **5→ 3 → 1 → 2 → 4**
  - **5→ 4 → 2 → 1 → 3**

For each figure/path:
Correct response 3 marks

(iii) **4 marks**

Column A Area of Land	ODD OR EVEN		
	Column B (Figure 3)	Column C (Figure 4)	Column D (Figure 5)
A	ODD (3)	**EVEN (2)**	**EVEN (2)**
B	ODD (3)	**EVEN (2)**	**EVEN (2)**
C	ODD (5)	**EVEN (2)**	**ODD (3)**
D	ODD (3)	**EVEN (2)**	**ODD (3)**

Each Correct Column 2 marks OR (whichever is the greater)
Each Correct Row 1 mark

(iv) **3 marks**

For an Euler path to exist, either of the following conditions must be met:

- **Every vertex (land mass) has an even number of edges (bridges) connected to it, or**
- **Exactly two vertices (land masses) have an odd number of edges (bridges) connected to them.**

Very good explanation - clear understanding demonstrated	3 marks
Fair explanation - limited understanding	2 marks

(b) **12 (8, 4) marks**

(i) **8 (5, 3) marks**

- **Abstraction is used by simplifying the real city map into a graph of points (land areas) and lines (bridges). Details like the shape of the river or distances are ignored because they are not important to the problem.**

Very good explanation - clear understanding demonstrated	5 marks
Fair explanation - limited understanding	3 marks

- **Important real-world details, such as distances or bridge widths, are ignored**
- **The abstract graph cannot show practical issues like time, safety, or walking difficulty**
- **Some real-world constraints may be missed because the model is oversimplified**
- **Any reasonable limitation**

Very good limitation - clear understanding demonstrated	3 marks
Fair limitation - limited understanding	2 marks

(ii) **4 marks**

Sequence:

- **A and B cross - 3 minutes**
- **A returns - 1 minute (total 4)**
- **C and D cross - 10 minutes (total 14)**
- **B returns - 3 minutes (total 17)**
- **A and B cross - 3 minutes (TOTAL OPTIMAL TIME 20 minutes)**

Very good explanation - clear understanding demonstrated	4 marks
Good explanation - clear information, lacking demonstration of full understanding	3 marks
Fair explanation - limited understanding	2 marks

(c) **10 marks**

Potential Positive Impacts

- **Enhanced access to navigation and route planning - users can find efficient routes more easily, particularly when travelling through unfamiliar areas.**
- **Reduced travel time and commuter stress - recommending quicker routes can reduce time spent in traffic and make journeys less stressful.**
- **Reduced traffic congestion - vehicles can be redirected away from busy routes, helping to reduce bottlenecks and improve traffic flow.**
- **Faster emergency service response times - optimal routing can help emergency services reach incidents more quickly, potentially saving lives and property.**
- **Lower fuel consumption and emissions - shorter and more efficient journeys can reduce fuel use, transport costs and environmental pollution.**
- **Improved transport, logistics and urban planning - route and traffic data can help improve public transport, delivery services and decisions about future infrastructure.**
- **Any other positive impact.**

Potential Negative Impacts

- **Privacy concerns from location tracking - collecting and storing users' location data may raise concerns about how personal information is used, stored or shared.**
- **Increased traffic through certain areas - optimised routes may divert vehicles through residential or culturally significant areas, increasing noise, congestion and disruption.**
- **Over-reliance on algorithms and loss of navigation skills - dependence on route-planning systems may reduce people's ability to navigate independently using maps or local knowledge.**
- **Unequal access and digital divide issues - people without smartphones, reliable internet access or digital skills may be unable to benefit equally from these systems.**
- **Implementation costs and risk of system failures - high costs, inaccurate data, software bugs or technical failures could result in poor route recommendations and travel disruption.**
- **Any other negative impact.**

For each impact (positive and negative):

Very good appraisal - clear understanding demonstrated	5 marks
Fair appraisal - limited understanding	4 marks

Question 15 **38 (14, 14, 10) marks**

(a) **14 (6, 4, 4) marks**

(i) **6 (2, 1, 2, 1) marks**

Digital sensors:

- **Door sensor (open or closed)**
- **Motion sensor (movement detected or not)**
- **Any valid response**

Analogue sensors:

- **Temperature sensor**
- **Humidity sensor**
- **Natural light sensor**
- **Any valid response**

For each digital sensor:
First correct 2 marks
Second correct 1 mark

For each analogue sensor:
First correct 2 marks
Second correct 1 mark

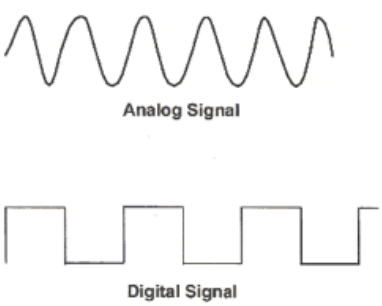
(ii) **4 (2, 2) marks**

- **Automatic window opener to control airflow**
- **Alarm or buzzer to warn of unsafe conditions**
- **Shade blinds to reduce light and heat**
- **Any valid response**

For each correct response 2 marks

(iii) **4 marks**

- **Digital sensors produce fixed outputs such as on/off or true/false values.**
- **Analogue sensors produce a range of values that change smoothly, such as temperature or humidity levels. This means analogue sensors can give more detailed information than digital sensors.**



Very good description - clear understanding demonstrated	4 marks
Fair description - limited understanding	2 marks

(b) **14 (6, 2, 6) marks**

(i) **6 (3, 3) marks**

Problem

- **If the temperature is above 30 and the door is open, the fan may turn ON and then immediately OFF, causing unstable or incorrect fan behaviour**

Solution

- **Use an `if-else` statement so the door condition is checked first, ensuring the fan stays OFF whenever the door is open**
- **Use an `and` operator to check if the temperature is above 30 and the state of the door**

For each part:

Very good description - clear understanding demonstrated	3 marks
Fair description - limited understanding	2 mark

(ii) **2 marks**

- **If humidity is too low, turn on the misting system**
- **If light level is too high, close the shade blinds**
- **If soil moisture is low, activate the irrigation system**
- **Any other valid rule that could be used to maintain healthy growing conditions in the greenhouse system**
- **Any reasonable response**

Very good description - clear understanding demonstrated	2 marks
Fair description - limited understanding	1 marks

(iii) **6 (3, 3) marks**

Benefit

- **Collecting this data allows the system to identify patterns and trends over time. For example, the company could see when temperatures are usually too high and adjust the fan rules to turn on earlier. This helps improve efficiency and maintain healthier growing conditions.**

Explain

- **Storing sensor readings and fan states makes it easier to detect faults or unusual behaviour in the system. For example, if the fan is often on but the temperature does not decrease, the system settings or equipment can be adjusted to improve performance.**

For each part:

Very good description - clear understanding demonstrated	3 marks
Fair description - limited understanding	2 marks

(c) **10 (4, 6) marks**

(i) **4 (2(1,1), 2(1,1)) marks**

Characteristic of Embedded Systems

- **Designed to perform a specific task**
- **Built into a larger system or device**
- **Operates with limited hardware resources**
- **Often works in real time**
- **Uses low power and runs continuously**
- **Fast response times**
- **Any other valid characteristic**

For each correct characteristic 1 mark

Justifications:

- **Specific task: The controller only needs to manage greenhouse conditions, which keeps the system simple and reliable.**
- **Part of a larger system: It must work with sensors and actuators like fans and heaters to control the greenhouse.**
- **Limited resources: Using minimal memory and processing power reduces cost and energy use.**
- **Real-time operation: The system must respond immediately to changes in temperature or humidity to protect plants.**

- **Low power use: The controller runs continuously, so low power consumption is important for efficiency.**
- **Fast response time: The greenhouse controller must react quickly to changes in temperature or humidity to maintain optimal growing conditions for plants.**
- **Low cost: A low-cost controller makes greenhouse automation more affordable for growers while still providing effective environmental control.**
- **Any valid justification for a named characteristic**

For each justification:
Good description - understanding demonstrated 1 mark

(ii) **6 (4,2) marks**

Any two from the following:

- **Unit testing: Individual parts of the system, such as the temperature sensor code or fan control rule, are tested to make sure each works correctly on its own.**
- **System testing: The full greenhouse system is tested together to check that sensors, rules, and outputs work properly under real greenhouse conditions.**
- **Integration testing: Checks that sensors, control rules, and outputs work correctly when combined.**
- **Acceptance testing: Ensures the system meets the horticulture company's requirements before final use.**
- **Regression testing: Confirms that new changes do not break existing features, such as fan or heater control.**
- **Functional Testing: Tests whether the system performs the functions it is supposed to perform (as per requirements).**
- **Non-Functional Testing: Tests how well the system performs (rather than what it does) e.g. performance, reliability, security etc.**
- **Any other valid stage of software testing e.g. release testing (alpha/beta), black/white box testing etc.**

For best response:
Very good response – clear understanding demonstrated 4 marks
Fair response – limited understanding 3 marks

For second best response:
Very good response – clear understanding demonstrated 2 marks
Fair response – limited understanding 1 mark

Section C — Programming — 80 marks

Question 16 — **80 (60, 20) marks**

(a) — **60 (10, 10, 10, 10, 5, 5, 10) marks**

Possible solution:

```
# Question 16 (a)
# Examination Number:

# Function to calculate readability score of text
def calculate_readability_score(text):

    # Split the text into a list of words
    words = text.split()
    print("The list of words in the text is:\n", words)

    # Initialise the word counters
    word_count = len(words) # Number of words in list

    short_word_count = 0
    # v - start
    for word in words:
        if len(word) <= 2:
            short_word_count += 1
    print("Short Words:", short_word_count)
    # v - end

    # vii - START
    complex_word_count = 0
    vowels = "AEIOUaeiou"
    for word in words:

        syllable_count = 0
        for ch in word:
            if ch in vowels:
                syllable_count = syllable_count + 1

        # subtract 1 for words ending in "e"
        last_char = word[len(word)-1]
        if last_char == "e":
            syllable_count = syllable_count - 1

        if syllable_count >= 3:
            complex_word_count += 1

    print("Complex Words:", complex_word_count)
    # vii - END

    # Initialise the number of sentences
    sentence_count = text.count(".") + text.count("!") # (iii)
```

```
45
46      # Calculate the readability score
47      #score = word_count * sentence_count
48      score = 0.4*(word_count/sentence_count + 100*((word_count-
    short_word_count)/word_count)) # vi
49      score = round(score, 2) # Round the score to 2 d.p.
50
51      return score
52
53  # Set the text for analysis
54  text1 = "Elaborate sentences influence readability in subtle and
    unpredictable ways."
55  text2 = "I do not like green eggs and ham! I do not like them Sam I am!"
56
57  # Calculate the score for text1
58  score1 = calculate_readability_score(text1)
59  print("Readability Score 1:", score1) # (i)
60  print() # (i)
61
62  # (ii) Calculate the score for text2
63  score2 = calculate_readability_score(text2) # (ii)
64  print("Readability Score 2:", score2) # (ii)
65  print() # (ii)
66
67  # (iv) - start
68  if score1 == score2:
69      print("Both scores are the same")
70  elif score1 < score2:
71      print("The first text is easier to read.")
72  else:
73      print("The second text is easier to read.")
74  # (iv) - end
75
```

(i) **10 marks (B-10 scale)**

10 marks	**Correct response** Correct implementation using solution above or similar.
7 marks	**Response with some merit** Any other reasonable attempt.

(ii) **10 marks (B-10 scale)**

10 marks	**Correct response** Correct implementation using solution above or similar.
7 marks	**Response with some merit** Any other reasonable attempt.

(iii) **10 marks (B-10 scale)**

10 marks	**Correct response** Correct implementation using solution above or similar.
7 marks	**Response with some merit** Any other reasonable attempt.

(iv) **10 marks (C-10 scale)**

10 marks	**Correct response** Correct implementation using solution above or similar.
7 marks	**Almost correct response** Correct implementation using solution above or similar but with minor syntax or semantic error.
3 marks	**Response with some merit** Any other reasonable attempt.

(v) **5 marks (C-5 scale)**

5 marks	**Correct response** Correct implementation using solution above or similar.
3 marks	**Almost correct response** Correct implementation using solution above or similar but with minor syntax or semantic error.
2 marks	**Response with some merit** Any other reasonable attempt.

(vi) **5 marks (C-5 scale)**

5 marks	**Correct response** Correct implementation using solution above or similar.
3 marks	**Almost correct response** Correct implementation using solution above or similar but with minor syntax or semantic error.
2 marks	**Response with some merit** Any other reasonable attempt.

(vii) **10 marks (D-10 scale)**

10 marks	**Correct response** Correct implementation using solution above or similar.
7 marks	**Almost correct response** Correct implementation using solution above or similar but with minor syntax or semantic error.
5 marks	**Response about half-right** Partially correct implementation using solution above or similar but with significant syntax or semantic error.
3 marks	**Response with some merit** Any other reasonable attempt.

(b) **20 marks**

Possible solution:

```
# Question 16 (b)
# Examination Number:

# Scrabble-like word scoring game

total_score = 0
high_score = 0
highest_scoring_word = ""

while True:

    # Prompt the user to enter a word
    word = input("Enter a word: ")

    # Ensure the word is letters only (A-Z or a-z)
    while word.isalpha() == False:
        print("Invalid word.")
        word = input("Enter a word: ")

    # Convert the word to upper case
    word = word.upper()

    # Logic to determine the score of the word
    score = 0
    for letter in word:
        if letter in "AEIOU":
            score += 1
        elif letter in "VWXYZ":
            score += 5
        else:
            score += 3

    # Update the total score so far
    total_score =  total_score + score

    # Display the word, its score and the total score so far
    print(f"Word: {word}, Score: {score}, Total score: {total_score}")

    # Keep tack of the highest scoring word
    if score > high_score:
        high_score = score
        highest_scoring_word = word

    # Play again if the user enters 'Y' or 'y'
    play_again = input(f"Play again? [Y/N]: ").upper()
    if play_again not in "Yy":
        break

# Display the final output message
print()
```

```
51 print(f"The highest scoring word was {highest_scoring_word}, {high_score}
   points")
52
```

	High level of achievement **All** of the following implemented correctly and efficiently	**Moderate level of achievement** Reasonable attempt to implement at least two of each of the following	**Low level of achievement** Poor attempt to complete any of the following
Program Inputs (5 marks)	• Word read in correctly • Play again read in correctly • Variable initialisation and use of assignment statements (5 marks)	• Word read in correctly • Play again read in correctly • Variable initialisation and use of assignment statements (4 marks)	• Word read in correctly • Play again read in correctly • Variable initialisation and use of assignment statements (3 marks)
Program Logic (Processing) (5 marks)	• Word converted to uppercase • Word validation • Logic to calculate and update score • Logic to maintain high score • Play again logic (5 marks)	• Word converted to uppercase • Word validation • Logic to calculate and update score • Logic to maintain high score • Play again logic (4 marks)	• Word converted to uppercase • Word validation • Logic to calculate and update score • Logic to maintain high score • Play again logic (3 marks)
Program Outputs (5 marks)	• Word, score and total score correctly displayed • High score displayed • Error message displayed (5 marks)	• Word, score and total score correctly displayed • High score displayed • Error message displayed (4 marks)	• Word, score and total score correctly displayed • High score displayed • Error message displayed (3 marks)
Programming Standards (5 marks)	• Program executes correctly with no syntax or runtime errors • Program meets all requirements • Program design is well explained with comments (5 marks)	• Program executes correctly with no syntax or runtime errors • Program meets more than half of the requirements • Program design is well explained with comments (4 marks)	• Program executes correctly with no syntax or runtime errors • Program meets any requirement • Program design is well explained with comments (3 marks)

Coursework (90 marks in total)	
The report	***Marks***
• Quality of report website structure and layout. • Evidence of adherence to the principles of good user interface design when creating the website. • Adherence to the word count and video length (penalties may apply).	**5**
1. Meeting the brief	
• Meeting the basic requirements of the brief. • Meeting the advanced requirements of the brief.	**27**
2. Investigation	
• Research on the brief, including research on existing solutions, systems or ideas that are aligned to the brief. • Consideration of how your research informs decisions about your project.	**10**
3. Plan and design	
• A clear, detailed description of the design objectives of the project. • Consideration of different options and justification of design choices. • Consideration of stakeholders and end user's needs. • The technologies that will be used. • System architecture/high-level flowchart showing components.	**18**
4. Create	
• Describe the key milestones of the development process. • Describe the testing that took place during the development process. • Explain a problem that was encountered in the development of the project and how it was overcome. • Describe the model or the embedded system algorithm that was programmed as part of the project.	**20**
5. Evaluation	
• An evaluation of your final artefact. • Suggest how you would further improve/iterate this project.	**10**
References and summary word count	
• You must also include references and/or a bibliography. • Include a summary of the word count of the report, including the total word count.	**0**

Higher grade	Ordinary grade	Reference Mark	Higher Mark	Ordinary Mark
1		81 - 90	81 - 90	90
2		72 - 80	72 - 80	90
3		63 - 71	63 - 71	90
4		54 - 62	54 - 62	90
5	1	45 - 53	45 - 53	81 - 90
6	2	36 - 44	36 - 44	72 - 80
7	3	27 - 35	27 - 35	63 - 71
8	4	23 - 26	23 - 26	54 - 62
	5	18 - 22	18 - 22	45 - 53
	6	14 - 17	14 - 17	36 - 44
	7	9 - 13	9 - 13	27 - 35
	8	0 - 8	0 - 8	0 - 26

COURSEWORK - conversion from reference mark to Ordinary-level mark

For Ordinary-level candidates, the final mark is found from the reference mark as follows:

- If the reference mark is 54 or more the final mark is 90.
- If the reference mark is at least 27 but less than 54, then add 36 to the reference mark to get the final mark.
- If the reference is at least 1 but less than 27, then double the reference mark and add 9 to get the final mark.
- If the reference mark is 0 the final mark is 0

Reference Mark	Conversion
54 or more	Award 90 marks
27 - 53	Add 36 marks
1 - 26	Multiply the reference mark by 2 and add 9 marks
0	0