Coimisiún na Scrúduithe Stáit
State Examinations Commission

Leaving Certificate 2025

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, 3, 5	0, 2, 3, 5	
10 mark scale				0, 3, 5, 7, 10
15 mark scale				0, 3, 7, 10, 15

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**

Output
5
10
20
2
1
False

Each correct response in order	1 mark
All correct values on 1 line	4 marks
Any two correct values in the proper sequence	2 marks

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

Memory Usage

- **ASCII is a fixed width encoding system that uses 7 bits allowing for 128 unique characters, which makes it very memory-efficient for English text. Extended ASCII uses 8 bits and therefore supports a maximum of 256 characters.**
- **Unicode may use 1 to 4 bytes per character, which can increase memory usage but allow over 1.1 million characters to be represented (including symbols from non-Roman character sets, mathematical symbols, emojis etc.).**

Compatibility

- **Unicode was designed to be backward compatible with ASCII (the first 128 characters in Unicode are identical to ASCII).**
- **Unicode offers far greater compatibility than ASCII. Because it supports characters from all languages, it enables applications in different countries to communicate seamlessly with one another.**

For each feature:

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

Question 3 **6 marks**

Ranking	Girl's Name	Project theme
1	**Kay**	Artificial intelligence
2	**Ada**	**Web development**
3	Grace	**Data analytics**
4	**Joy**	**Embedded system**

Each correct response 1 mark

Question 4 **6(3,3) marks**

- **Resource Management refers to how the operating system controls and coordinates the computer's hardware resources such as CPU time, memory (RAM), secondary storage and input/output devices.**
- **File management refers to how the operating system organises and keeps track of files on storage devices (like hard drives or USB drives) and ensures they can be created, accessed or modified when needed.**

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

Question 5 **6 marks**

Red: 0A in hex = 10 in decimal
Green: 33 in hex = $\mathbf{3 \times 16^1 + 3 \times 16^0 = 51}$
Blue: F0 in hex = $\mathbf{15 \times 16^1 + 0 \times 16^0 = 240}$

Answer: RGB(10, 51, 240)

For each conversion:
Correct response: 2 marks
Response with some merit 1 mark

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

- **Improved accuracy. AI systems can detect patterns that may be missed by human doctors, potentially leading to more accurate diagnoses.**
- **Faster diagnosis. AI can quickly analyse medical images, test results, and patient records, reducing the time it takes to reach a diagnosis.**
- **Early detection. AI can help identify diseases such as cancers or neurological conditions at an earlier stage thus improving the chances of successful treatment.**
- **Reduced workload for doctors. By handling routine diagnostic tasks, AI allows medical professionals to focus on more complex or urgent cases.**
- **Availability. AI systems can be available 24 hours a day 7 days a week making medical care more available to patients. They can also be used in areas with limited access to specialists.**
- **Any other valid benefit described.**

Limitations

- **Lack of human judgement. AI cannot fully understand patient emotions which can be essential in making well-rounded clinical decisions.**
- **Possible lack of accuracy. If AI systems are trained on biased or incomplete data, they can produce inaccurate or unfair results, especially for underrepresented groups. Also, if new diseases emerge after the AI is trained, it might not have sufficient information to make a correct diagnosis.**
- **Ethical concerns regarding patient privacy, data security and accountability. AI systems require access to large amounts of patient data, raising concerns about how this sensitive information is stored and used. Also, it can often be difficult to understand how AI systems arrive at a diagnosis, which can reduce trust and accountability.**
- **High costs. Developing, and maintaining advanced AI systems can be expensive and require significant technical infrastructure and expertise.**
- **Any other valid limitation described.**

For each part:

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

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

(a)

Algorithm 1: Linear search
Algorithm 2: Binary search

Each correct response: 1 mark

(b)

- **Linear search may be more efficient as there is no sort overhead.**
  Also, target item will sometimes be the first item (or one of the first items) in the list.
  For relatively small lists this can be more efficient than a binary search.

OR

- **In general, binary search is more efficient as at worst it just requires log(n) operations to complete its task. Linear search on the other hand requires O(n) operations.**

Very good justification - clear understanding demonstrated	4 marks
Fair justification – limited understanding	2 mark

Question 8 **6 marks**

- **Internet Protocol (IP) is the general protocol used for sending all types of data in the form of packets over the internet (e.g. browsing websites and sending emails).**
- **Voice over Internet Protocol (VoIP) is a specific application of IP used for voice and video communication (e.g. Zoom and WhatsApp).**

Very good distinction - clear understanding demonstrated	6 marks
Good distinction - clear information, lacking demonstration of full understanding	4 marks
Fair distinction – limited understanding	2 mark

Award 1 mark each for correct protocol name only.

Question 9 **6(2,4) marks**

(a)

Any two syntax errors from the following:

- **Line 1. Incorrect case - For instead of `for`**
- **Line 3. Missing closing bracket**
- **Line 5. Missing opening quote**
- **Line 8. Missing colon after `else`**
- **Line 9. Variable `number` is undefined**

Each syntax error 1 mark

(b)

- **Logic error: The program never displays *FizzBuzz* as intended. This should happen when the loop counter variable, `i` is 15.**
- **Suggested solution: Move the test for divisibility by both 3 and 5 (i.e. line 6) to the start of the if-statement (i.e. before the individual tests).**

Logic error correctly identified 2 marks
Suggested solution 2 mark

Question 10 **6(2,2,2) marks**

Examples	Limitations
Any two from the following: • **Customer support chatbots** • **Online shopping recommendations** • **Self-driving cars** • **Medical diagnosis (or monitoring)** • **Smart home systems** • **Banking transactions (e.g. payments)** • **Payroll systems** • **School/college administration systems** • **Any other valid example**	• **High development/maintenance cost** • **Job displacement** • **Errors in data can lead to mistakes in outcomes.** • **Security (vulnerable to cyberattacks)** • **Ethical concerns (e.g. privacy, fairness, accountability)** • **Limited human understanding** • **Any other valid limitation**

For each example 2 marks
Limitation:
Very good explanation - clear understanding demonstrated 2 marks

Question 11 **6(3,3) marks**

Advantage

- **Voters receive content relevant to their interests.**
- **Helps voters make informed decisions.**
- **Encourages greater engagement potentially leading to higher turnouts.**
- **Political parties can focus their resources on reaching the right audience, saving time and money.**
- **Any other valid advantage.**

Potential Risks

- **Users may only see content that agrees with their views, limiting exposure to other perspectives (echo chambers/filter bubbles).**
- **Algorithms may spread false or misleading information tailored to a user's preferences.**
- **Privacy concerns.**
- **Any other valid potential risk.**

For each advantage/disadvantage	
Very good explanation - clear understanding demonstrated	3 marks
Fair explanation – limited understanding	2 marks

Question 12 **6 (1,2,1,2) marks**

- **Arithmetic Logic Unit (ALU): Performs arithmetic operations (e.g., addition, subtraction) and logical operations (e.g., AND, OR, NOT) necessary for processing instructions.**
- **Control Unit (CU): Directs and coordinates the activities of the CPU by fetching, decoding, and executing instructions from memory. It ensures all CPU components work together efficiently.**
- **Registers: Provide small, high-speed storage locations within the CPU to temporarily hold data, instructions, or addresses during processing.**
- **Cache: Stores frequently accessed data and instructions close to the CPU to reduce latency and improve processing speed.**
- **Program Counter (PC): Keeps track of the memory address of the next instruction to be executed, ensuring the CPU processes instructions in the correct sequence.**
- **Clock: Generates a consistent timing signal (or "clock pulse") that synchronizes the operations of the CPU. It ensures that instructions are executed in a controlled, sequential manner by determining the speed at which the CPU processes data.**

For each correctly identified CPU component	1 mark
For each purpose:	
Very good explanation - clear understanding demonstrated	2 marks
Fair explanation – limited understanding	1 mark

Section B — Long Questions — 76 marks

Answer any **two** questions.

Question 13 — **38 (12, 15, 11) marks**

(a) — **12 (6, 2, 4) marks**

(i) — **6 (3 (1, 1, 1), 3 (1, 1, 1)) marks**

Roles and Responsibilities:

- **System architect: Defines the overall structure of the application (system components, protocols, etc.), decides on technology stack, promotes technologies.**
- **Analyst: Identifies and collaborates with stakeholders to identify and document the system requirements, develops system use cases.**
- **Designer: Designs the application e.g. pseudo-code, flowcharts, data flow diagrams, wireframes, implements prototypes, database design, produces design documents**
- **Software developer: Writes the application code, implementation, unit testing, debugging.**
- **Software tester: Tests the application code – unit test, system test, functional/non-functional testing, documents and tracks test results and status/progress.**
- **UI/UX Designer: Create a user-friendly user interface for the application.**
- **IT Support Technician: Provides ongoing technical support to resolve any technical issues faced by users of the system.**
- **Any other valid role and associated responsibilities.**

For each role correctly identified 1 mark
For each responsibility:
Good explanation - clear understanding demonstrated 1 mark

(ii) — **2 (1, 1) marks**

Any two from the following:

- **Waterfall/Staged.**
- **Agile/Iterative.**
- **Any valid software development process/methodology.**

For each correct response 1 mark

(iii) **4 (2, 2) marks**

Option 1: Waterfall/Staged

- **Highly structured, follows a sequence of phases e.g. analysis, design, implementation, deployment and maintenance.**
- **Each phase must be completed before moving to the next, ensuring a structured and predictable process.**
- **Comprehensive documentation such as requirements specifications, design plans, test cases.**
- **Suitable for large projects when requirements are unlikely to change.**

Option 2: Agile/Iterative

- **Less structured/more flexible, breaks project into smaller iterations called sprints.**
- **Functional components are delivered incrementally.**
- **Customer collaboration throughout the project lifecycle.**
- **Very responsive to dynamic/changing requirements.**

For each feature:
Very good justification - clear understanding demonstrated 2 marks
Fair justification – limited understanding 1 mark

(b) **15 (4, 1, 4, 6) marks**

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

Store Staff

- **Goal: To have a system that quickly registers returned bottles and updates stock. System should be reliable and easy to use. System should be highly/fully automated and software should give feedback on errors / alerts when something goes wrong - e.g. machine full. System should result in less waste.**
- **Benefit To provide fast service to customers and reduce workload.**

Customer

- **Goal to be able to scan returned bottles and receive an immediate refund.**
- **Benefit is receipt of refund and contribution to cleaner environment.**

For each goal
Very good explanation - clear understanding demonstrated 1 mark

For each benefit
Very good explanation - clear understanding demonstrated 1 mark

(ii) **1 mark**

Any from the following:

- **Store manager**
- **Government regulator**
- **Administrator**
- **Maintenance technician**
- **Logistics manager**
- **Environmental analyst**
- **Any other valid stakeholder**

Valid stakeholder identified 1 mark

(iii) **4 (2, 2) marks**

- **Machine Details: This database would store machine details e.g. location, date installed, date last emptied, status etc.**
- **Container Tracking: This database system could use unique identifiers for each container to keep a track of when they enter the system and possibly a recycling or reuse status for each container.**
- **Finance Management: This database could be used to store deposit amounts paid by consumers and records the refunds given when containers are returned. It could be used to generate transaction and financial reports.**
- **Any other valid database management use case.**

For each use case suggested:
Very good explanation - clear understanding demonstrated 2 marks
Fair explanation – limited understanding 1 mark

(iv) **6 (3, 3) marks**

Any two of the following:

- **Automated image recognition for bottle scanning.**
- **Demand prediction for collection logistics.**
- **Predictive maintenance for return machines.**
- **Personalised incentives to engage users.**
- **Data analysis (e.g. for environmental impact reporting).**
- **Any other valid way AI could be used in the DRS application e.g. chatbot feature.**

For each suggestion
Very good explanation - clear understanding demonstrated 3 marks
Fair explanation – limited understanding 2 marks

(c) **11 (6, 5) marks**

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

- **Screen Reader: A screen reader can help visually impaired users by reading out text on the system interface, such as instructions or transaction details, so they can participate in the deposit return scheme.**
- **Voice Recognition Software: This technology can allow users with mobility impairments to interact with the DRS system by speaking commands, such as "return container" or "check balance," without needing to use a keyboard or mouse.**
- **Braille Display: For users who are blind or have severe vision impairment, a Braille display can provide tactile feedback of the system's text, allowing them to read information like instructions or container details in Braille.**
- **Simplified User Interface: A simplified interface with larger buttons, clearer fonts, and high-contrast colours can assist users with cognitive disabilities or low vision in navigating the DRS system more easily.**
- **Any other valid assistive technology for a visually impaired user.**

For each assistive technology correctly identified	1 mark
For each explanation:	
Very good explanation - clear understanding demonstrated	2 marks
Fair explanation – limited understanding	1 mark

(ii) **5 marks**

Unit Testing: Focuses on testing individual components of a program carried out by developers as they write their code.

Functional Testing: Tests the overall functionality of the software, ensuring all features work together as expected from the user's perspective. It is usually done later in the development process.

Very good distinction - clear understanding demonstrated	5 marks
Good distinction - clear information, lacking demonstration of full understanding	3 marks
Fair distinction – limited understanding	1 mark

Question 14 **38 (10, 15, 13) marks**

(a) **10 (2, 4, 4) marks**

(i) **2 marks**

- **To introduce (define) the function `gcd`.**

Correct response 2 marks

(ii) **4 marks**

a	b
45	18
27	**18**
9	**18**
9	**9**

For each correct row 1 mark
Full correct response 4 marks

(iii) **4 marks**

- **The number of iterations is not known in advance of running the algorithm.**
- **The number of iterations required can vary depending on the arguments passed into the function `gcd`.**

Very good explanation - clear understanding demonstrated 4 marks
Fair explanation – limited understanding 2 marks

(b) **15 (4, 6, 5) marks**

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

Base Case

- **the condition that stops the recursive function from calling itself indefinitely (or until a stack overflow error occurs).**
- **Each recursive call works towards the base case decomposing the problem into simpler problems until the simplest version of the problem is reached. The answer to the simplest version of the problem is known e.g. `recursive_gcd(a, 0)` is `a`.**
- **In the example code the condition `b == 0` is the base case.**

.

Recursive Case

- **the part of the function that calls itself to solve a simpler/smaller version of the problem.**
- **eventually the recursive calls converge on the base case.**
- **In the example code the recursive case is on line 6 i..e. `recursive_gcd(b, a % b)`.**

For each case:

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

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

The variable `b`:

- **This is the first argument (parameter/value) passed into the function as `a`.**
  (*Within* the function `b` determines whether the base case (`b == 0`) is reached).

The expression `a%b`:

- **This is the second argument (parameter/value) passed into the function as `b`.**
  It computes the remainder when `a` is divided by `b`. The remainder becomes the new value of `b` in the next recursive call.
  (*Within* the function its use is to reduce the problem size by applying the principle that the GCD of two numbers (`a` and `b`) is the same as the GCD of `b` and the remainder of `a % b`).

For each answer:

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

(iii) **5 (1, 4) marks**

- **gcd**

Correct response: 1 mark

- **The non-recursive solution is more suitable for finding the GCD of very large numbers on machines with limited RAM because it avoids the potentially high memory overhead of recursion (and the possibility of causing a stack overflow).**
- **In a non-recursive solution memory usage remains constant, as the loop reuses the same variables regardless of the size of the numbers or the number of iterations.**
- **In a recursive solution the memory usage grows with each recursive call as the call stack increases. For very large numbers this could exhaust the stack leading to a stack overflow error.**

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

(c) **13 (2, 5, 2, 4) marks**

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

- **Left sub-list: [10]**
- **Right sub-list: [29, 14, 37]**

For each correct sub-list 1 mark

(ii) **5 marks**

- **After the first pass, quicksort continues by applying the same process to each sub-list created during partitioning. If a sub-list has only one element, it is already sorted. For larger sub-lists, a new pivot is chosen, and the list is divided into elements less than the pivot and elements greater than or equal to it. This divide-and-conquer approach is repeated recursively on each sub-list. Once all sub-lists are reduced to single elements, they are combined in order, resulting in a fully sorted list.**
- **Appropriate diagram.**

Very good explanation - clear understanding demonstrated	5 marks
Good explanation - clear information, lacking demonstration of full understanding	3 marks
Fair explanation - limited understanding	1 mark

(iii) **2 (1, 1) marks**

Pivot selection strategies

- **First element**
- **Last element**
- **Middle**
- **Median of first, middle and last element**
- **Median of medians**
- **Random**

For each correct strategy 1 mark

(iv) **4 (2, 2) marks**

Best case:

- **The best-case time complexity of quicksort is $O(n\ log\ n)$ because, in this scenario, each pivot divides the list evenly into two halves. This results in a balanced recursion tree with about $log\ n$ levels. At each level, the algorithm processes all n elements once during partitioning, so the total work done across all levels is $n\ \times\ log\ n$.**

Worst case:

- **The worst-case time complexity of quicksort is $O(n^2)$. This happens when the pivot repeatedly creates very unbalanced splits, such as when it is always the smallest or largest element in the list. Instead of dividing the list in half, one sub list ends up with n - 1 elements and the other with 0, creating a recursion tree with n levels. Since the algorithm still processes n elements at each level, the total number of operations becomes $n \times n = O(n^2)$. This makes quicksort inefficient in the worst case, especially on already sorted or reverse-sorted lists if no precautions are taken.**

For each case (best and worst):

Very good explanation - clear understanding demonstrated 2 marks

Fair explanation – limited understanding 1 mark

Question 15 **38 (11, 16, 11) marks**

(a) **11 (1, 2, 4, 4) marks**

(i) **1 mark**

Output variable:

- **TEST OUTCOME**

Correct response 1 mark

(ii) **2 (1, 1) marks**

Input variables: Any two from

- **GENDER**
- **AGE**
- **TRANSMISSION**
- **INSTRUCTOR EXPERIENCE (Years)**

For each correct response 1 mark

(iii) **4 (2, 2) marks**

Any two from the following:

- **Younger drivers (under 20) tend to have higher failure rates.**
- **Drivers using automatic cars tend to pass, while manual transmission drivers have higher failure rates.**
- **More experienced instructors are correlated with higher pass rates.**
- **The number of females who pass is the same as the number of males who fail.**
- **Any other valid pattern.**

For each pattern correctly identified 2 marks

(iv) **4 (2, 2) marks**

Any two from the following:

- **Number of lessons.**
- **Number of previous attempts.**
- **Driving test time of day.**
- **Weather conditions.**
- **Familiarity with car.**
- **Any other valid variable.**

Each variable correctly identified 2 marks

(b) **16 (4, 4, 8) marks**

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

Limitations

- **Models are abstractions that simplify complex systems. By design, they omit many details to focus on specific aspects of the system being studied. This makes them inherently "wrong" in representing the full complexity of reality.**
- **Underlying assumptions of model may not always hold true.**
- **Data limitations (e.g. bias, errors, size) can make models inaccurate.**
- **Ethical concerns regarding privacy, security and accountability.**
- **High development costs.**
- **Any other valid limitation.**

Benefits

- **Models make complex systems easier to understand, identify patterns, relationships, and areas for intervention.**
- **Models can provide valuable predictions within a defined scope.**
- **Models help decision-makers weigh options and test scenarios in a controlled, low-risk environment e.g. a simulation of traffic flow can guide city planners in designing more efficient road systems.**
- **Any other valid benefit.**

For each part:

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

(ii) **4 marks**

- **Abstraction is important when developing a model because it allows us to focus on the most essential elements of a system while ignoring unnecessary details.**
- **By simplifying complex systems, abstraction makes it easier to understand, analyse, and simulate the system's behaviour.**
- **Without abstraction, models would become too complicated, difficult to interpret, and less effective in predicting outcomes or testing different scenarios.**

Very good explanation - clear understanding demonstrated	4 marks
Fair explanation – limited understanding	2 marks

(iii) **8 (4, 4) marks**

- **Education: Modelling can be used to simulate different teaching methods or learning environments, helping educators identify the most effective approaches for improving student engagement and performance. For example, a model could simulate student interactions in a virtual classroom to optimise teaching strategies.**
- **Mental Wellbeing: Modelling mental wellbeing can involve simulating various factors like stress levels, social interactions, and lifestyle choices to predict outcomes and suggest personalised interventions. For instance, a model could help identify triggers for anxiety and recommend coping strategies, improving mental health management.**
- **Environment: Environmental models simulate natural systems, such as climate change or ecosystem dynamics, to predict future conditions. These simulations help in understanding the potential impacts of human activity, guiding policy decisions and strategies for sustainable resource management.**
- **Traffic Management: Traffic flow simulations can model the movement of vehicles in a city to identify congestion points and optimise traffic signal timings. This helps improve traffic efficiency, reduce delays, and lower accident rates, ultimately enhancing overall transportation systems.**

For each area:

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

(c) **11 (3, 2, 6) marks**

(i) **3 marks**

- **Each student and teacher could be modelled as an agent with attributes (e.g., vaccination, infection rate etc.). Agents could be made to interact in different locations (e.g., classroom, canteen etc) and the model rule could be used to simulate infection spread (e.g., contact duration, mask-wearing etc).**

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

(ii) **2 marks**

Any one from the following:

- **Clusters of infection forming in areas with high interaction, like hallways.**
- **Clusters of infection associated with specific activity e.g. sports, musical etc.**
- **Patterns showing higher infection rates among specific people/age groups (superspreaders).**
- **Recovery can lead to patterns such as slowing transmission rates or the eventual end of an outbreak.**
- **Any other valid emergent behaviour.**

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

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

Strategies - any one from the following:

- **Measures to reduce crowding e.g. social distancing/ staggered lunch breaks/quarantine.**
- **Mask wearing.**
- **Vaccination/booster campaigns.**
- **Regular testing / contact tracing.**
- **Increased hygiene.**
- **Enhanced air ventilation systems.**
- **Any other valid strategy.**

Correct strategy identified	2 marks

Measure success:

- **Compare infection rates before and after the strategy.**
- **Analyse the time it takes for the disease to stop spreading.**

For each response:

Very good explanation - clear understanding demonstrated	4 marks
Fair explanation – limited understanding	2 marks

Section C — Programming — 80 marks

Question 16 — **80 (50, 30) marks**

(a) — **50 (5, 5, 5, 5, 5, 10, 15) marks**

Possible solution:

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

def get_grade(result):
    grade = "Unsuccessful"

    if result >= 80:
        grade = "Distinction"
    elif result >= 65:
        grade = "Upper Merit"
    # part iii - start
    elif result >= 50:
        grade = "Lower Merit"
    elif result >= 40:
        grade = "Pass"
    # part iii - end

    return grade

# Calculate and display the mean of a list of results
results = [39,32,62,88,51,62,64,81,77] # Initialise the list
N = len(results) #initialise N to the number of results
total = 0 #initialise the running total to 0

# Loop N times
for i in range(N):
    total = total + results[i] # running total

# Divide by the total number of results to give the mean
arithmetic_mean = total/N # part ii
arithmetic_mean = round(arithmetic_mean, 2) # part i
# Display the answer
print("The mean percentage mark is", arithmetic_mean)

# part (iv) - start
grade = get_grade(arithmetic_mean)
print("The grade for the average result is", grade)
# part (iv) - end

# part (v) - start
highest = max(results)
lowest = min(results)
print("The lowest score is", lowest)
print("The highest score is", highest)
# part (v) - end

# part (vi) - start
a = 0 # count of results less than 40
b = 0 # count of results between 50 and 79 inclusive
for result in results:
    if result < 40:
        a += 1
```

```
    elif result >= 50 and result <= 79:
        b += 1
print("The number of scores below 40 is", a)
print("The number of scores between 50 and 79 inclusive is", b)
# part (vi) - end

# part (vii) - start
longest_run = []
current_run = [results[0]]

for i in range(1, N):
    if results[i] > results[i - 1]:
        current_run.append(results[i])
    else:
        if len(current_run) > len(longest_run):
            longest_run = current_run
        current_run = [results[i]]

# Check one last time at the end of the loop
if len(current_run) > len(longest_run):
    longest_run = current_run

print("Longest run of result increases is", longest_run)
# part (vii) - end

```

(i) **5 marks (B-5 scale)**

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

(ii) **5 marks (B-5 scale)**

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

(iii) **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.

(iv) **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.

(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) **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.

(vii) **15 marks (D-15 scale)**

15 marks	**Correct response** Correct implementation using solution above or similar.
10 marks	**Almost correct response** Correct implementation using solution above or similar but with minor syntax or semantic error.
7 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) **30 marks**

Possible solution:

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

# Initialise the list
nums = [27, 13, 32, 50, 16]

# Display the list
print("The initial list of values is:", nums)

# Sort the list
nums.sort()

# Display the sorted list
print("The sorted list of values is:", nums)

# Determine the median
N = len(nums)
if N == 0:
    print("The list is empty. Cannot compute the median.")
else:
    if N % 2 != 0:
        median = nums[N//2]
    else:
        median = (nums[N//2-1] + nums[N//2])/2

    # Display the median
    print("The median is", median)

```

	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)	• Initial list of integers correctly initialised • List sorted correctly • Variable initialisation and use of assignment statements (5 marks)	• Initial list of integers correctly initialised • List sorted correctly • Variable initialisation and use of assignment statements (4 marks)	• Initial list of integers correctly initialised • List sorted correctly • Variable initialisation and use of assignment statements (3 marks)
Program Logic (Processing) **(10 marks)**	• Length of list calculated • Logic to test even/odd • Calculation of median (even) • Calculation of median (odd) • Logic to test empty list (10 marks)	• Length of list calculated • Logic to test even/odd • Calculation of median (even) • Calculation of median (odd) • Logic to test empty list (7 marks)	• Length of list calculated • Logic to test even/odd • Calculation of median (even) • Calculation of median (odd) • Logic to test empty list (5 marks)
Program Outputs (5 marks)	• Initial list of integers displayed • Sorted list of integers displayed • Median displayed • Error message displayed (5 marks)	• Initial list of integers displayed • Sorted list of integers displayed • Median displayed • Error message displayed (4 marks)	• Initial list of integers displayed • Sorted list of integers displayed • Median displayed • Error message displayed (3 marks)
Programming Standards (10 marks)	• Program executes correctly with no syntax or runtime errors • Program meets requirements • Program design is well explained with comments • Meaningful variable/function names (10 marks)	• Program executes correctly with no syntax or runtime errors • Program meets requirements • Program design is well explained with comments • Meaningful variable/function names (7 marks)	• Program executes correctly with no syntax or runtime errors • Program meets requirements • Program design is well explained with comments • Meaningful variable/function names (5 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 (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 into existing interactive information systems. • Research into the chosen area or topic that will be analysed through the information system.	**10**
3. Plan and design	
• A clear, detailed description of the project and how it will meet the requirements. • A description of the technologies you will use and their role within your project. • A flowchart diagram to show how the project will work.	**15**
4. Create	
• A progress log covering the key milestones of the development process. • Evidence of testing applied during development. • Explain a problem that was encountered in the development of the project and how it was overcome. • An explanation of an algorithm that you designed that has been used for the analytics of your project.	**25**
5. Evaluation	
• An evaluation of your project based on the requirements set out in the brief. • Suggest how you would further improve/iterate this project.	**8**
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