



Coimisiún na Scrúduithe Stáit
State Examinations Commission

Leaving Certificate 2023

Marking Scheme

Computer Science

Ordinary 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. Scales labelled B divide candidate responses into four categories (correct response, almost correct response, partially correct response, 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
No. of categories	3	4	5
5 mark scale	0, 2, 5	0, 2, 3, 5	
10 mark scale		0, 5, 8, 10	0, 3, 5, 7, 10
15 mark scale		0, 5, 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 (3 categories)

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

B-scales (4 categories)

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

C-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**

Column A Print statement	Column B True or False
<code>print(a == b)</code>	False
<code>print(b != a)</code>	True
<code>print(a > b)</code>	True
<code>print(b <= a)</code>	True
<code>print(a == 10)</code>	False
<code>print(b != 21)</code>	True

Each correct item

1 mark

Question 2**4 + 2 marks****(a)**

- General-purpose computers can have many uses, embedded systems have a limited or specific number of uses
- General-purpose computers can be easily reprogrammed, embedded systems can't easily be upgraded
- General-purpose computers are generally higher cost, embedded systems are relatively low cost

Good description – clear understanding demonstrated 4 marks

Fair description – limited understanding

2 mark

(b)

- Examples of Embedded Systems can be any digital device created for specific tasks – these include domestic appliances such as washing machines, dish washers, etc.
- Do not accept Micro:bit, Raspberry Pi as valid examples

Valid example 2 marks

Question 3

6 marks

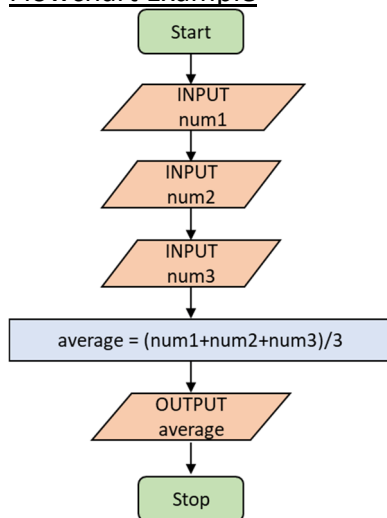
Pseudocode Example

```
input num1
input num2
input num3
```

```
average = (num1 + num2 + num3) / 3
```

```
print average
```

Flowchart Example



Candidates can use either pseudocode or a flowchart, as long the process can be followed/understood.

Attempt to input number(s)	1 mark	OR	3 separate numbers input	2 marks
Attempt at calculating average	1 mark	OR	average calculated correctly	2 marks
Print / output of average	1 mark			
All steps in correct logical order	1 mark			

Question 4

6 marks

14/2 = 7 Remainder 0
7/2 = 3 Remainder 1
3/2 = 1 Remainder 1
1/2 = 0 Remainder 1

OR

14 = 8 + 4 + 2 + 0
1 1 1 0

Ans: 1110

Ans: 1110

Correct answer	6 marks
Half correct conversion	4 marks
Any valid conversion	2 marks

Question 5**6 marks**

- **Bicycle** – pattern recognition could be used to identify common characteristics of a bike such as the shape, 2 wheels, handlebars.
- **Human** - pattern recognition could be used to identify common characteristics of a human such as the shape, 2 legs, 2 arms, height.
- **Car** - pattern recognition could be used to identify common characteristics of a car such as the shape, 4 wheels, speed of movement.
- **Other relevant examples & explanations can be used – not just what is shown in Figure 1. E.g. Facial recognition on mobile phones, finger print recognition on finger print scanners.**

Valid example and very good description 6 marks
 Valid example and fair description 4 marks
 Valid example **OR** fair description 2 marks

Question 6**6 marks**

Column A Variable Description	Column B Data Type
Member Name	String
Member Age	Integer
Amount Paid	Float
Most Popular Days to Visit	List
Member Completed Induction – Yes/No	Boolean

First correct item 2 marks
 Additional correct items 1 mark

Question 7**6 marks**

- **Process data and instructions and to control the rest of the computer system**
- **The CPU is often compared to the brain of the computer and as such deals with all of the inputs, processes and outputs of a computer system**
- **The CPU processes all data on a computer system and carries out the fetch, decode, execute cycle**

Very good explanation - clear understanding demonstrated 6 marks
 Good explanation - clear information, lacking full understanding 4 marks
 Fair explanation - limited understanding 2 mark

Question 8**2+4 marks****(a)****Dara**

Correct answer

2 marks

(b)

- Binary search looks at the middle item of a list & checks if that is the item it is searching for. If the item is not found it then it splits the list in half, meaning there are less items to search. This process is repeated until item is found or there is nothing left to search.
- Binary search works by dividing the list into two halves, eliminating one half and conquering the other half by repeating this process.

Good explanation – clear understanding demonstrated 4 marks

Fair explanation – limited understanding 2 mark

Question 9**6 marks**

Column A Example of Data	Column B Discrete or Continuous
The number of players in a football squad	Discrete
The number of students in a class	Discrete
The speed of a school bus	Continuous
The temperature of a classroom	Continuous
The number of days it rained in March	Discrete
The weight of a box	Continuous

Each correct item

1 mark

Question 10**6 marks**

Complete the truth table for the OR logic gate, shown below in **Figure 3**.

A	B	out (A OR B)
0	0	0
0	1	1
1	0	1
1	1	1

Accept False (for 0) and True (for 1); Do not accept either A or B as a valid response

1 correct answer	2 mark
2 correct answers	3 marks
3 correct answers	4 marks
All correct answers	6 marks

Question 11**2+2+2marks****(a)****Not Graded**

Correct answer 2 marks

(b)**Distinction**

Correct answer 2 marks

(c)

- **Converts (casts) the text entered to an integer**
- **Ensures that the datatype of the variable percentage will be an integer**

Good explanation – clear understanding demonstrated	2 marks
Fair explanation – limited understanding	1 mark

Question 12**6 marks**

- **Braille keyboard** - allows the user to type and enter text or instructions for the computer in Braille
- **Screen reader** - software that enables people with severe visual impairments to use a computer
- **Screen magnifier** - software that provides enlarged screen content for visual impairments
- **Sip-and-puff** - device that allows users to enter commands into a computer system via mouth
- **Accept any other relevant examples and explanations such as,**
 - **Large monitors.**
 - **Alternative types of keyboard and mouse (e.g. on-screen keyboards, keyboard with modifications, etc.).**
 - **Accessibility tools built into popular OS, browsers, and application software (e.g. software to adjust screen colours, font characteristics, magnification applications, keyboard filters, etc.).**
 - **Computers with voice output.**
 - **Word prediction software.**
 - **Adapted pointing/input devices (e.g. gesture devices, eyeball tracking devices, chin pointing devices etc.).**
 - **Wands and sticks.**
 - **Joysticks.**
 - **Trackballs.**
 - **Touch screens.**
 - **Spellchecker**

For each example:

Valid example

1 mark

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 (14, 12, 12) marks****(a)****14 (4, 10) marks****(i)****4 (1,1,1,1) marks**

Column A Definition	Column B Term
A data structure that can store a collection of elements of the same data type	Array
A character that represents a mathematical or logical action	Operator
A command for handling a decision	Conditional
A memory location that can store a value	Variable

Each correct item

1 mark

(ii)**10 (2,2,2,2,2) marks**

	i	j	while i < j	print(i)
1	1	10	True	1
2	2	9	True	2
3	3	8	True	3
4	4	7	True	4
5	5	6	True	5
6	6	5	False	6 or blank

Fully correct row

2 marks

or

Row with 2 or more correct items 1 mark

(b) 12 (3, 3, 3, 3) marks

(i) €5

Correct answer 3 marks

(ii) €3

Correct answer 3 marks

(iii) €3

Correct answer 3 marks

(iv) €2

Correct answer 3 marks

(c) 12 (6, 6) marks

(i) 6 marks

Car 1 & 2 charge for 30 minutes - total running time 30 minutes
Car 2 comes out and Car 3 enters - total running time 30 minutes
After 30 minutes Car 2 replaces Car 1 - total running time 60 minutes (Car 1 full)
After 30 minutes Car 2 & Car 3 full - total running time 90 minutes
Ans: 1.5 hours (90 minutes)

Correct answer with workings 6 marks

Correct answer only 4 marks

Work of merit 2 marks

(ii) 6 marks

- **Algorithmic thinking** – Algorithmic thinking is when you consider all of the information within the problem, plan out the plan out the steps towards coming up with a solution and try to find the most effective way to the end goal.
- **Decomposition, abstraction, pattern recognition, generalisation, logical thinking and evaluation** are all considered computational thinking skills.

Valid computational thinking skill named 2 marks

Good explanation – clear understanding demonstrated 4 marks

Fair explanation – limited understanding 2 mark

Question 14**38 (14, 12, 12) marks****(a)****14 (6, 8) marks****(i)****6 (4, 2) marks****Stakeholders**

- **Example stakeholders: Irish Government, commercial investors, App development team, RTE staff/management/board, people who use RTE services, advertisers etc.**

For each stakeholder identified

2 marks (max. 4 marks)

End Users

- **Members of the public who stream RTE services online or use the app**
- **Other relevant end users can be accepted as long as an explanation given**

End user identified

2 marks

(ii)**8 (4, 4) marks**

- **Role 1: Tester**
Description 1: Tests the software at different stages throughout the process to look for errors.
- **Role 2: Project Manager**
Description 2: Makes sure that all of the team members know what they need to do, also makes sure everything is running on time.

There are many other roles / terms within software development such as Product Owner, Business Analyst, System Analyst, Designer, Developer/ Programmer/ Coder. Accept any relevant role and description.

Note: Where two separate roles are considered equivalent/to have the same meaning, marks are awarded for the best response only. Zero is awarded for the other response.

Apply marks for each (2) role:

Valid role identified

1 mark

Good description – clear understanding demonstrated

3 marks

Fair description – limited understanding

2 marks

(b) 12 (6, 6) marks
(i) 6 (3, 3) marks

- **Create**

- Writing the code to implement the design
- Ongoing testing to ensure the code is working
- Creating databases for storing data
- Creating any physical elements of the project
- Accept any other valid tasks that take place within the create stage.

Good description – clear understanding demonstrated 3 marks
Fair description – limited understanding 2 mark

- **Document**

- Documentation of system requirements for software
- Creation of user guide document
- Creation of user maintenance document
- Creation of frequently asked questions
- Accept any other valid tasks that take place within the document stage.

Good description – clear understanding demonstrated 3 marks
Fair description – limited understanding 2 mark

(ii) 6 (3, 3) marks

- Flexibility – short sprint-scrum cycles
- Software is produced early on
- Software is developed in fast cycles
- Each version builds on previous
- Each version is thoroughly tested before release
- Good for small, time-critical projects
- User involvement throughout
- Limited planning is needed to get started

Each (2) valid advantage 3 marks

(c) 12 (6, 6) marks

(i) 6 marks

- Unit testing is when small/specific sections of code are tested by the developer while it is being created
- Unit tests can be carried out on functions / modules of code

Very good explanation - clear understanding demonstrated 6 marks
Fair explanation - limited understanding 4 marks

(ii)

6 (3, 3) marks

- **Advantages**

- **Real end users try it out**
- **Tested by public before release**
- **Feedback improves product quality**
- **Chance to find errors/ flaws/ issues before release**
- **Any other valid advantage**

Valid advantage

3 marks

- **Disadvantages**

- **Users can leak information about software before it is released**
- **Process can be time consuming and delay the release**
- **Can be a waste of time / money if the Beta testers do not use the software properly**
- **Beta testers might not be representative of actual end users**
- **Any other valid disadvantage**

Valid disadvantage

3 marks

Question 15

38 (14, 12, 12) marks

(a)

14 (10, 4) marks

(i)

10 (5, 5) marks

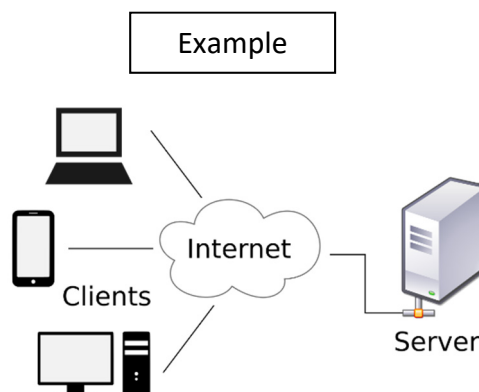
- User accounts setup with secure passwords
- Restricted accounts so not all users have the same privileges
- Firewall to monitor all data moving in and out of the network
- Use of VPN
- Server is kept in a secure location
- All data is up to date and backed up
- Antivirus/ anti-spyware software installed on the network
- Network policy in place for all users
- Accept any other relevant security measure for a network

Each (2) valid recommendation

5 marks

(ii)

4 marks



- Client-server model works by servers constantly 'listening for requests from clients. As requests are received they are processed by the server which responds by sending the result back to the client in the form of a response.

Clear diagram and explanation

4 marks

Clear diagram of client-server model without explanation

3 marks

Clear explanation of client-server model without diagram

3 marks

Some relevant information, not complete

2 marks

(b) **12 (6, 6) marks**
• Advantages **6 (3, 3) marks**

- **A centralised system can be more efficient**
- **Faster/ Less time spent on accessing information**
- **Less money spent on printing and postage**
- **Easier for citizens to access / update their information**
- **Services are more accessible in regard to mobility and/or location**
- **Accept any other relevant point**

Each (2) valid advantage 3 marks

• Disadvantages **6 (3, 3) marks**

- **System could be hacked, and data stolen**
- **If the internet goes down the system isn't usable**
- **Digital divide - people with no access or skills to use technology are disadvantaged**
- **Governments have too much information / control / invasion of privacy**
- **Potential job losses with more reliance on technology**

Each (2) valid disadvantage 3 marks

(c) **12 marks**

For each benefit:

Name: 1 mark

Explanation:

Very good explanation - clear understanding demonstrated 5 marks

Fair explanation - limited understanding 2 mark

For each danger:

Name: 1 mark

Explanation:

Very good explanation - clear understanding demonstrated 5 marks

Fair explanation - limited understanding 2 mark

Section C

Programming

80 marks

Question 16

80 marks

(a)

50 (15, 10, 10, 5, 5, 5) marks

Possible solution:

```

1 # Question 16(a)
2 # Examination Number:
3 print("*****") #(ii)
4 print("Times Table program") #(i)
5 print("*****") #(ii)
6
7 number = int(input("Enter number: ")) #(iii)
8 if number < 0: #(iv)
9     print("This program does not support negative numbers")
10 else: #(iv)
11
12     for i in range(13): #(v)
13         print(i, "x", number, "=", number*i) #(vi)
14

```

(i)

15 marks (B-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.
5 marks	Response with some merit Any other reasonable attempt.

(ii)

10 marks (B-10 scale)

10 marks	Correct response Correct insertion of row of asterisks in both locations
8 marks	Almost correct response Correct implementation using solution above or similar but with minor syntax or semantic error
5 marks	Response with some merit Any other reasonable attempt.

(iii)

10 marks (C-10 scale)

10 marks	Correct response Correct implementation using solution above or similar. Note variable does not have to be cast to <code>int</code> in the <code>input</code> statement for full marks
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.

(iv)

5 marks (B-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 (A-5 scale)

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

(vi)

5 marks (B-5 scale)

5 marks	Correct response Correct implementation using solution above or similar.
3 marks	Almost correct response Almost correct implementation using solution above or similar but with minor syntax error.
2 marks	Response with some merit Any attempt at string concatenation, <code>format</code> function or similar

(b)

30 marks

Possible solution:

```
1 # Question 16(b)
2 # Examination Number:
3
4 # Display a welcome message
5 print("Welcome to Temperature Alert System")
6
7 # Ask the user to enter the temperature
8 temperature = int(input("Enter temperature value in degrees: "))
9
10 # Display a message based on the value of temperature
11 # If temperature is less than 20 degrees display too cold
12 if temperature <20:
13     print("Too cold. Turn up heating.")
14 # Else if temperature is between 20 and 24 display just right
15 elif temperature>=20 and temperature <=24:
16     print("Temperature is just right.")
17 # Otherwise temperature must be above 24 degrees. Display too warm.
18 else:
19     print("Too warm. Turn down heating. ")
```

	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
Programming Standards (5 marks)	• Program executes with no syntax or runtime errors • Program meets requirements • Program design is well explained with comments • Meaningful variable/function names (5 marks)	• Program executes with no syntax or runtime errors • Program meets requirements • Program design is well explained with comments • Meaningful variable/function names (4 marks)	• Program executes with no syntax or runtime errors • Program meets requirements • Program design is well explained with comments • Meaningful variable/function names (3 marks)
Program Inputs (5 marks)	• User correctly prompted to enter a temperature value in degrees Celsius • Temperature value converted to integer • Variable initialisation and use of assignment statements (5 marks)	• User correctly prompted to enter a temperature value in degrees Celsius • Temperature value converted to integer • Variable initialisation and use of assignment statements (4 marks)	• User correctly prompted to enter a temperature value in degrees Celsius • Temperature value converted to integer • Variable initialisation and use of assignment statements (3 mark)
Program Logic (Processing) (10 marks)	• Conditional Logic • Use of relational/Boolean operators (10 marks)	• Conditional Logic • Use of relational/Boolean operators (7 marks)	• Conditional Logic • Use of relational/Boolean operators (5 marks)
Program Outputs (10 marks)	• Initial welcome message • Output message 1: <i>Too cold. Turn up heating</i> • Output message 2: <i>Temperature is just right</i> • Output message 3: <i>Too warm. Turn down heating</i> (10 marks)	• Initial welcome message • Output message 1: <i>Too cold. Turn up heating</i> • Output message 2: <i>Temperature is just right</i> • Output message 3: <i>Too warm. Turn down heating</i> (7 marks)	• Initial welcome message • Output message 1: <i>Too cold. Turn up heating</i> • Output message 2: <i>Temperature is just right</i> • Output message 3: <i>Too warm. Turn down heating</i> (5 marks)

Coursework (90 marks in total)	
1. Meeting the Brief	Marks
- Meeting the basic requirements of the brief. - Meeting the advanced requirements of the brief.	27
2. Investigation and Plan	
- Research into the context of the brief, existing solutions and initial thoughts on your own project. - A detailed description of the chosen project and the objectives.	14
3. Design	
- A clear detailed description of how the project will be developed. - A flowchart diagram to show how the project will work. - A description of how abstraction and modelling will be applied.	14
4. Implementation	
- An overview covering the key milestones of the development process. - Explain a problem that was encountered in the development of the project and how it was overcome. - An explanation of a piece of code or algorithm that was important in the development of the project.	14
5. Testing	
- An overview of the testing carried out. - A detailed test case table.	11
6. Evaluation	
- An evaluation of your project based on your initial design and your project objectives. - 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

