G.C.E. (Advanced Level) Examination - 2026
20 - Information & Communication Technology
Model Marking Scheme

English Medium | Papers I and II | Examiner edition

Version 2.0 - 9 November 2026

Independent model. Prepared for the supplied 2026 question paper. This is not an official Department of Examinations marking scheme. All detailed allocations and concessions below are proposed model rules.

General marking instructions

Award marks for the evidence specified in each component. Underlined words identify the essential idea; equivalent wording is accepted unless exact labels, identifiers or wording are required.

Alternatives separated by a slash are acceptable substitutes. Award each component once only. A correct answer together with a contradictory answer does not earn the same identification mark.

Where a component states “requires A”, award it only when component A is earned. Other components are independent. Do not import dependency rules from an unrelated question in a previous year.

For Python and Arduino code, the syntax, case and indentation relevant to the component must be correct. Accept equivalent functioning code subject to the specified method. Ignore SQL keyword case. Preserve supplied identifier spelling where the answer must fit existing code.

Do not award more than the maximum for a subpart. Do not deduct marks for an extra explanation unless it contradicts the credited answer. Do not require material that the question does not ask for.

Mark all attempted Part B questions and count the best four. Retain fractional marks throughout. If using the reporting convention in the references, round upward only the final Paper II total.

Record Paper I and Paper II separately. This scheme specifies no combined weighting or grade boundaries.

Paper / section	Maximum
Paper I: 50 multiple-choice items	50
Paper II, Part A: Questions 1-4, all compulsory	40
Paper II, Part B: any four of Questions 5-10	60
Paper II total	100

Notes: Read the moderation notes on the next page before marking. The companion Review and Moderation Notes gives the source comparison, complete MCQ reasoning and the basis for each disputed item.

Moderation notes for use with this model

Apply the following rules consistently to the entire group. They are model decisions, not official concessions. If the paper has not yet been administered, correct the affected questions first.

Reference	Working rule
Paper I 17, 18	Use intended options 5 and 3. Item 17 uses a simplified RSA-style private/public-key description; item 18 assumes Internet access from a private IPv4 LAN.
Paper I 29	Intended answer: 5, counting constraints on the proposed tuple. The SQL statements are malformed. For an uncorrected sitting, this model awards the item to all candidates, as for 30 and 38.
Paper I 30	Literal DO1 (letter O) finds no rows; no listed option is correct. Award 1 to all candidates for the uncorrected item. Corrected D01 gives option 1.
Paper I 38	Literal new list is invalid Python; no listed option is correct. Award 1 to all candidates for the uncorrected item. Corrected new_list gives option 1.
Paper II 5(b)	Use the complement on (a′b + ab′) only, with c outside it. Credit the six algebra components shown. If a candidate reads the long bar as including c and disproves that reading, apply the stated concession.
Paper II 7(b)(ii)	The printed allocation is missing. Allocate 1 mark, giving Question 7 a total of 15. This is inferred.
Paper II 7(b)(iii)	Likely intended answer: ITA. Since the scenario does not establish relative autonomy, also accept SDA or “cannot be determined”. Do not require an explanation for this one-mark identification question.
Paper II 8(a)	The bracketed terms name attributes only. Accept sensible entity and relationship names; retain supplied attribute names.
Paper II 9	Main answer follows the explicit 12-day example. Also credit a consistently applied progressive interpretation as specified alongside the answer. Do not penalise the same interpretation again in (b) and (d).

Score control

Question	1	2	3	4	5	6	7	8	9	10
Maximum	10	10	10	10	15	15	15	15	15	15

Paper II = Q1 + Q2 + Q3 + Q4 + the best four scores from Q5-Q10. Maximum 100. For the uncorrected Paper I under this model, the maximum remains 50: 47 answer-key marks plus the three neutralised items.

Paper I - Answer key

Q1-10	Answer	Q11-20	Answer	Q21-30	Answer	Q31-40	Answer	Q41-50	Answer
1	3	11	1	21	3	31	2	41	4
2	4	12	1	22	1	32	2	42	1
3	1	13	1	23	3	33	1	43	2
4	1	14	4	24	1	34	2	44	5
5	4	15	2	25	3	35	4	45	1
6	1	16	4	26	5	36	3	46	5
7	2	17	5	27	2	37	3	47	2
8	2	18	3	28	5	38	ALL*	48	1
9	4	19	5	29	ALL*	39	4	49	3
10	5	20	4	30	ALL*	40	5	50	3

1 mark per item. Maximum: 50 marks.

Notes: ALL* means award 1 mark to every candidate for the uncorrected item, under the model moderation policy. Where the paper is corrected before use, replace Q29, Q30 and Q38 by options 5, 1 and 1 respectively. Do not apply both treatments.

For the other 47 questions, award 1 only for the keyed option. No response or multiple uncancelled responses earns 0. No negative marking.

Corrections needed before a fresh sitting

Q	Correction	Key after correction
29	Use syntactically valid INSERT statements with a column list and VALUES. Keep the proposed values and constraints unchanged.	5
30	Change DeptID = 'DO1' to DeptID = 'D01'.	1
38	Change print(mark_list, new list) to print(mark_list, new_list).	1

The companion review includes the rationale for all 50 answers. The compact key here is intended for marking scripts.

Paper II - Part A
Question 1

1(a)(i) HTML replacements [4]

Blank	1	2	3	4	5	6	7	8
Option	E	H	C	A	F	I	O	G

A. 0.5 for each correct option label, in its correct position. **4**

Notes: Exact option labels are required by the question. Repeated occurrences of the same numbered blank earn credit once only. Keywords: action, method, label, text, select, option, date, submit.

1(a)(ii) CSS syntax correction [0.5]

Missing colon after **background-color**. Correct declaration: `background-color: blue;`

A. Identifies the missing colon and gives the correct declaration. **0.5**

1(a)(iii) Effect on display [0.5]

The browser ignores the invalid declaration; the submit button does not receive the specified blue background from this rule.

A. Correct effect on the submit button. **0.5**

Notes: Do not accept “the whole page fails to display”. Other valid declarations still apply.

1(b) PHP replacements [2]

```
1: bookid       2: date       3: $var1       4: $var2
```

A. 0.5 for each correct replacement in order. **2**

Notes: Exact spelling and case are needed to match the supplied names and variables. The quotes around the first two blanks are already supplied.

1(c)(i) Browser output [2]

Library Book Records

Book ID	Category	Title	Status
B101	Science	Mechanics	Available
B102		Chemistry Basics	Borrowed
B103	ICT	Web Design	Available
B104		Programming Basics	Borrowed

A. Heading and all four column headers. **0.5**

B. Four correct book records in order. **0.5**

C. Science category cell spans the first two records. **0.5**

D. ICT category cell spans the last two records. **0.5**

Question 1 (continued) / Question 2

1(c)(ii) Additional book record [1]

```
<tr><td>B105</td><td>CSS Basics</td>
    <td>Available</td></tr>
```

A. Correct three-cell row after B104, before .	**0.5**
B. Changes the ICT cell in the B103 row to rowspan="3".	**0.5**

Notes: Do not add another category cell to the B105 row.

Total for Question 1: 10 marks

2(a)(i) Initiators [1]

LibMIS: student. SchoolLIB: librarian.

A. Student for LibMIS.	**0.5**
B. Librarian for SchoolLIB.	**0.5**

2(a)(ii) Operational bottleneck [1]

The librarian manually checks the paper student record before registering each student.

A. Identifies manual paper-record verification.	**1**

2(a)(iii) Data-verification gap [1]

A. LibMIS automatically validates against the central school database.	**0.5**
B. SchoolLIB manually checks paper-based student records.	**0.5**

2(a)(iv) Required process modification [1]

Allow students to initiate registration through the online page instead of requiring the librarian to enter each registration.

A. Student self-registration replacing librarian-only entry.	**1**

Notes: A verification change alone does not answer this part.

2(a)(v) Reengineered process [1]

Student submits details online → automatic verification against the central database → account creation → email notification.

A. Student self-service submission.	**0.5**
B. Automatic validation and account creation for valid details.	**0.5**

Question 2 (continued) / Question 3

2(a)(vi) DFD labels [1]

P: Registration details; Q: Student details;
R: User account details; S: Register user.

A. All four correct: 1; two or three correct: 0.5; otherwise: 0. **1**

Notes: Use terms from the supplied list. A repeated answer earns credit only in its correct position.

2(a)(vii) Testing technique [2]

Black-box testing: checks input/output behaviour without knowledge of the internal implementation.

A. Black-box testing. **1**

B. Explains external behaviour testing without inspecting internal code; requires A. **1**

2(b)(i) Secure payment mechanism [1]

Third-party payment service / third-party payment processor / intermediary digital wallet.

A. Correct intermediary payment-service type. **1**

Notes: PayPal alone repeats the example and does not name the type.

2(b)(ii) E-business transaction [1]

G2C / Government-to-Citizen.

A. G2C. Also accept C2G under the direction-neutral model rule. **1**

Notes: Do not accept B2C. The transaction is between government and citizen.

Total for Question 2: 10 marks

3(a)(i) Output [1]

```
25
```

A. Correct output only. **1**

3(a)(ii) Purpose [1]

Finds and displays the largest non-negative integer entered before the first negative input.

A. Maximum of the eligible inputs; negative sentinel excluded. **1**

3(a)(iii) Purpose of found [1]

Records whether at least one non-negative input was processed, to distinguish valid data from no eligible input.

A. Correct flag purpose. **1**

Notes: Zero is valid because the loop condition is n >= 0. The printed “positive numbers” message is imprecise.

Question 3 (continued)

3(b) Average-program replacements [3]

```
1: 0
2: total + n
3: count + 1
4: n = int(input())
5: count
6: total / count
```

A. 0.5 for each correct replacement in its correct position. **3**

Notes: Accept equivalent expressions fitting the surrounding code. Blank 4 is a complete input statement. The negative sentinel is not included in the total or count.

3(c)(i) Flowchart replacements [2]

```
1: reverse
2: n > 0
3: n % 10
4: reverse * 10 + digit
5: n = n // 10
```

A. Blanks 1 and 2 both correct. **0.5**

B. Blank 3 correct. **0.5**

C. Blank 4 correct. **0.5**

D. Blank 5 correct. **0.5**

Notes: MOD and DIV are acceptable in the flowchart. The symbol “= 0” is already printed after blank 1. Accept an equivalent terminating condition for the stated positive-integer input.

3(c)(ii) Python implementation [2]

```
n = int(input())
reverse = 0
while n > 0:
    digit = n % 10
    reverse = reverse * 10 + digit
    n = n // 10
print(reverse)
```

A. Integer input and initialization of reverse. **0.5**

B. Correct while condition and terminating integer-division update. **0.5**

C. Correct digit extraction and reverse accumulation. **0.5**

D. Outputs the accumulated reverse after the loop. **0.5**

Notes: A while loop implementing the given arithmetic algorithm is required. A string slice alone does not implement the flowchart. Correct indentation and syntax are needed for the credited component.

Total for Question 3: 10 marks

Question 4

4(a)(i) File block chains [2]

File	Chain	Marks
1	20 → 22 → -1	0.5
2	21 → 24 → 25 → -1	0.5
3	23 → 28 → 26 → 27 → -1	1

A. File 1 chain. **0.5**
B. File 2 chain. **0.5**
C. File 3 chain. **1**

Notes: The -1 marker is not a data block. Accept chains ending at their last block if termination is clear.

4(a)(ii) Combined disk space [1]

(2 + 4) × 4 KB = 24 KB.

A. Six blocks / correct substitution. **0.5**
B. 24 KB. **0.5**

4(a)(iii) Allocation method [2]

Indexed allocation: an index gives direct access to individual blocks and allows non-contiguous allocation as the file grows.

A. Indexed allocation. **1**
B. Explains direct/random block access without traversing a linked chain; requires A. **1**

4(b)(i) State saved on P3 → P2 switch [1]

Save P3's execution context (program counter/register contents/status) in P3's PCB.

A. Execution context / next-instruction address. **0.5**
B. Stored in P3's PCB. **0.5**

4(b)(ii) Context-switch overhead [1]

Saving/restoring state and switching processes takes CPU time without executing useful user-process work.

A. Correct explanation of overhead. **1**

4(c)(i) Maximum virtual address space [0.5]

2^{48} bytes.

A. Correct power of two and byte unit. **0.5**

4(c)(ii) Frame-number bits [0.5]

$2^{33} / 2^{12} = 2^{21}$ frames → 21 bits.

A. 21 bits. **0.5**

Question 4 (continued) / Part B - Question 5

4(c)(iii) Physical address [1]

Offset = 8193 mod 4096 = 1.
Physical address = 2 × 4096 + 1 = 8193.

A. Correct offset of 1. **0.5**

B. Correct physical address of 8193. **0.5**

4(d) Device driver [1]

The driver provides the printer-specific communication/control interface, translating the OS's I/O requests into commands understood by the printer.

A. Connects the driver to hardware-specific OS communication. **1**

Total for Question 4: 10 marks

Part B: answer any four questions. Each question carries 15 marks.

5(a) De Morgan's theorems [3]

```
(AB)' = A' + B'
(A+B)' = A'B'
```

A	B	(AB)'	A' + B'	(A+B)'	A'B'
0	0	1	1	1	1
0	1	1	1	0	0
1	0	1	1	0	0
1	1	0	0	0	0

A. First theorem stated. **0.5**

B. Second theorem stated. **0.5**

C. Complete proof of the first theorem. **1**

D. Complete proof of the second theorem. **1**

Notes: A correct combined truth table proves both theorems. Accept valid algebraic proofs. Equivalent complement notation is acceptable.

5(b) Boolean-algebra proof [6]

```
L = a'b'c + a'bc' + ab'c' + abc
  = (a'bc' + ab'c') + (a'b'c + abc)
  = (a'b + ab')c' + (a'b' + ab)c

Let X = a'b + ab'.
X' = (a + b')(a' + b)
   = aa' + ab + a'b' + b'b
   = ab + a'b'

Therefore L = Xc' + X'c
            = (a'b + ab')c' + (a'b + ab')'c
```

Question 5 (continued)

5(b) Allocation [6]

A. Regroups minterms into the c′ and c groups.	**1**
B. Factors to (a′b + ab′)c′ + (a′b′ + ab)c.	**1**
C. Uses De Morgan to obtain X′ = (a + b′)(a′ + b).	**1**
D. Distributes the product correctly.	**1**
E. Eliminates complementary products to obtain X′ = ab + a′b′.	**1**
F. Substitutes and reaches the required Xc′ + X′c identity.	**1**

Notes: Accept another valid algebraic sequence for the same six stages. For each stage, the algebra must be valid; a truth table alone is not the requested method. If the bar is read as including c, a correct counterexample (a=b=c=0 gives 0 ≠ 1) earns the full 6 under this model concession. Do not add this to derivation marks.

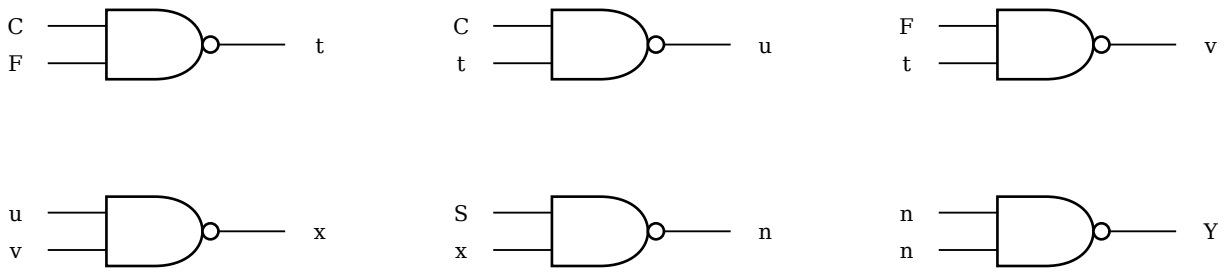
5(c) Access-control circuit [6]

Active-high inputs: S = supervisor key, C = clerk key, F = valid supervisor fingerprint. Y = 1 opens the door.

S	C	F	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Y = SC′F + SCF′ = S(C XOR F)

NAND-only circuit



Identical wire labels denote electrical connections. All six gates are NAND.

A. Clearly defined input/output variables and active levels.	**1**
B. Truth table: 0.5 per fully correct pair of rows, in the order shown.	**2**
C. Correct Boolean expression.	**1**
D. Correct NAND XOR stage producing x = C XOR F.	**1**
E. Correct NAND enable/inversion producing Y = Sx.	**1**

Notes: Equivalent NAND-only designs are accepted. The all-three-active input must give 0. Do not require this exact gate layout. For the table, row pairs are 000/001, 010/011, 100/101, 110/111.

Total for Question 5: 15 marks

Question 6

6(a) Purpose of NAT [2]

Translates private internal IPv4 addresses to public addresses, enabling the private network to communicate over the Internet.

A. Private/public address translation. **1**

B. Internet access or sharing/conservation of public addresses. **1**

6(b) Total address count [2]

$2^{32-16} = 2^{16} =$ 65,536 addresses.

A. Correct exponent / method. **1**

B. 65,536; do not subtract network and broadcast addresses. **1**

Notes: As this part asks for the total only, a correct final count earns both marks even without working.

6(c) Subnet allocation [5]

Department	Network address	Subnet mask	Usable range	Broadcast
Production	192.168.0.0/17	255.255.128.0	192.168.0.1 to 192.168.127.254	192.168.127.255
Marketing	192.168.128.0/19	255.255.224.0	192.168.128.1 to 192.168.159.254	192.168.159.255
Human Resource Management	192.168.160.0/19	255.255.224.0	192.168.160.1 to 192.168.191.254	192.168.191.255
Accounting	192.168.192.0/19	255.255.224.0	192.168.192.1 to 192.168.223.254	192.168.223.255
IT	192.168.224.0/19	255.255.224.0	192.168.224.1 to 192.168.255.254	192.168.255.255

A. 1 for each fully correct row, including all four requested fields. **5**

Notes: Accounting and IT may exchange their /19 networks. The original three allocations remain fixed. No partial credit within a row. The two new networks divide the remaining 192.168.192.0/18 equally.

6(d) Inter-building communication media [2]

Fibre-optic cable; point-to-point wireless / microwave.

A. One suitable medium. **1**

B. A second distinct suitable medium, or an explicit suitable link for the other new building. **1**

Notes: A fibre link to each new building earns 2. A single unallocated medium name earns 1. Accept UTP where the candidate specifies a compliant distance (normally up to 100 m per Ethernet channel). No justification is required for the standard fibre/wireless answers.

6(e) Firewall location and justification [4]

Internet / ISP router ↔ Firewall ↔ internal LAN / departmental subnets.

A. At the boundary between the Internet connection and internal network. **2**

B. Filters incoming/outgoing traffic according to security rules. **1**

C. All departments are protected because external traffic passes through it. **1**

Notes: An integrated router/firewall at the boundary is acceptable. A firewall protecting only one endpoint does not satisfy the network-wide placement requirement.

Total for Question 6: 15 marks

Question 7 - Embedded system

7(a)(i) Arduino blanks [4]

```
A: 2       B: INPUT       C: HIGH       D: HIGH
```

A. 1 for each correct blank. **4**

Notes: Pin 2 follows the PIR OUT wire in the supplied diagram. Require exact symbolic constants or their correct equivalent values: INPUT = 0, HIGH = 1. Do not confuse the PIR pin with LED pin 8.

7(a)(ii) Buzzer with the LED [2]

```
void loop() {
    int motion = digitalRead(pirPin);
    if (motion == HIGH) {
        digitalWrite(ledPin, HIGH);
        tone(9, 1000);
        delay(10000);
        noTone(9);
        digitalWrite(ledPin, LOW);
    }
}
```

A. Starts an audible tone on pin 9 when motion is HIGH. **1**

B. Stops the buzzer with the LED after 10 seconds. **1**

Notes: Accept tone(9, frequency, 10000) with the LED's existing delay and switch-off. Any suitable audible frequency earns credit. HIGH/LOW alone needs an explicitly assumed active buzzer; a passive piezo requires an alternating signal.

7(a)(iii) Night detection [2]

LDR / light sensor, connected to A0 / another analog input pin.

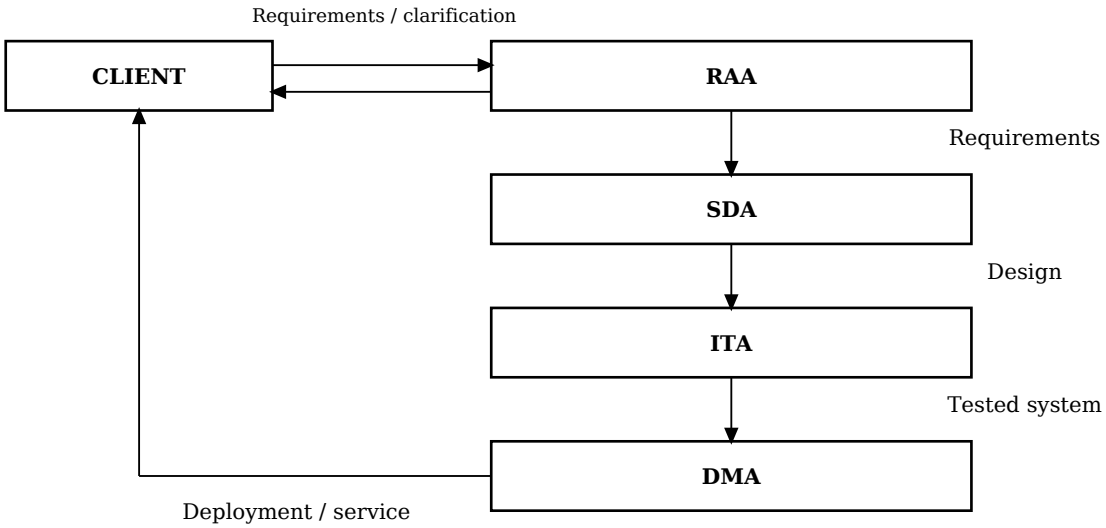
A. Suitable light sensor. **1**

B. Suitable analog input pin for that sensor. **1**

Notes: The question asks only for the sensor and pin. Do not require a circuit drawing, resistor value, threshold or program. A bare LDR would normally be wired as a voltage divider.

Question 7 (continued)

7(b)(i) Multi-agent workflow [2]



A. All four agents shown in waterfall order: RAA → SDA → ITA → DMA. **1**

B. Client interaction and correctly directed/labelled handoffs; requires A. **1**

Notes: Shapes may vary. Accept reasonable feedback paths and client approval interactions. A diagram is required.

7(b)(ii) Interface agent [1]

RAA (Requirements Analysis Agent).

A. RAA; also accept an agent clearly given the interface role in a valid diagram. **1**

Notes: The 1-mark allocation is inferred: it is missing in the question paper. No explanation is required by the question.

7(b)(iii) Lower self-autonomy [1]

ITA is the likely intended answer.

A. ITA / SDA / cannot be determined, under the moderation rule. **1**

Notes: The scenario supplies role names, not constraints establishing relative autonomy. Accept one of these responses; no supporting reason is required for this identification item.

7(c)(i) Bit and qubit [1]

A. A classical bit represents one of two states, 0 or 1. **0.5**

B. A qubit can be in a superposition of the 0 and 1 basis states. **0.5**

Notes: Formal amplitude notation is unnecessary. “Stores every possible answer at once” without the superposition idea is insufficient.

7(c)(ii) Public-key security challenge [2]

A. Quantum algorithms can make certain hard problems underlying public-key systems tractable, e.g. factoring or discrete logarithms. **1**

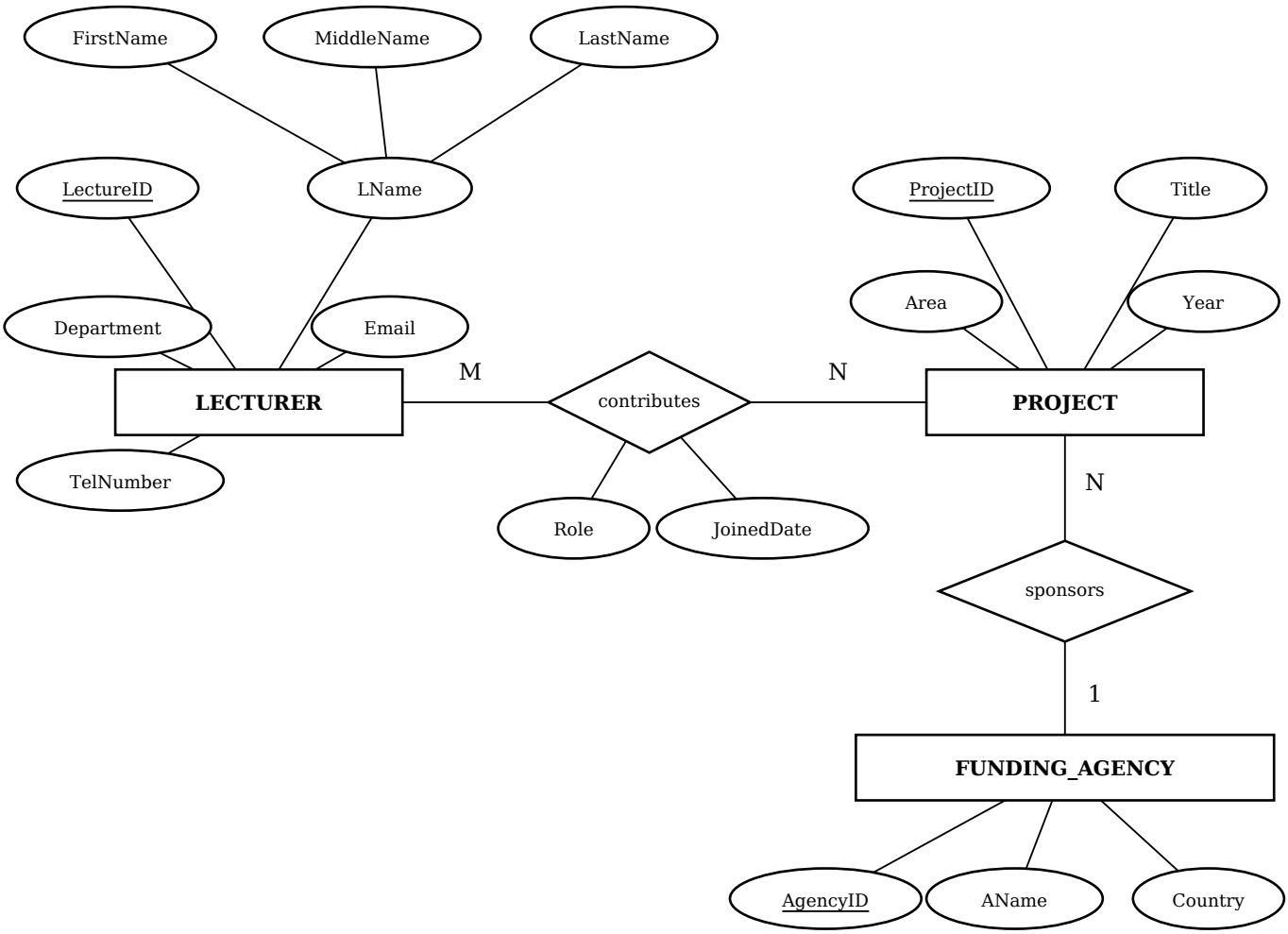
B. This can allow recovery of secret keys / decryption / compromise of signatures. **1**

Notes: Accept a syllabus-level explanation of solving the underlying mathematical problem much faster and compromising the key. Naming Shor’s algorithm is not required. Do not accept “all encryption is instantly broken”.

Total for Question 7: 15 marks

Question 8 - Database design

8(a)(i) Entity-relationship diagram [4]



A. LECTURER with all attributes and underlined LectureID.	**0.5**
B. PROJECT with all attributes and underlined ProjectID.	**0.5**
C. FUNDING_AGENCY with all attributes and underlined AgencyID.	**0.5**
D. LName shown as composite: FirstName, MiddleName, LastName.	**0.5**
E. Lecturer-project M:N relationship.	**0.5**
F. Role and JoinedDate attached to that relationship.	**0.5**
G. Agency-project 1:N relationship.	**0.5**
H. Each project assigned to exactly one agency (notation or explicit statement).	**0.5**

Each project has exactly one funding agency. Lecturer-project participation may be optional; an agency may sponsor many projects.

Notes: Accept equivalent standard ER notation and sensible entity/relationship names. Keep the printed attribute spelling LectureID. Do not require a multi-valued TelNumber: the scenario does not specify multiple telephone numbers.

Question 8 (continued)

8(a)(ii) Relational schema [4]

LECTURER(LectureID, FirstName, MiddleName, LastName,
Department, Email, TelNumber)
PROJECT(ProjectID, Title, Area, Year, AgencyID)
FUNDING_AGENCY(AgencyID, AName, Country)
CONTRIBUTES(LectureID, ProjectID, Role, JoinedDate)

```
PROJECT.AgencyID       → FUNDING_AGENCY.AgencyID
CONTRIBUTES.LectureID  → LECTURER.LectureID
CONTRIBUTES.ProjectID  → PROJECT.ProjectID
```

A. Four correct table/attribute sets: 0.5 each.	**2**
B. All four primary keys correctly underlined (both attributes for CONTRIBUTES).	**1**
C. All three foreign-key arrows point to the referenced primary keys.	**1**

Notes: The separate arrow lines above show the exact endpoints. Equivalent arrows on the schema are accepted. Do not require the composite LName as an extra stored column.

8(b)(i) Two partial dependencies [2]

```
StudentID → StudentName
ClubID    → ClubName
```

A. StudentID → StudentName, or another valid partial dependency.	**1**
B. A second distinct valid partial dependency.	**1**

Notes: ClubID → AdvisorID is also accepted. ClubID → AdvisorName is a valid derived partial dependency under the stated club/advisor assumptions. Both StudentID and ClubID are proper subsets of the composite membership key.

8(b)(ii) Transitive dependency [1]

```
(StudentID, ClubID) → AdvisorID → AdvisorName
```

A. Correct transitive chain, or AdvisorID → AdvisorName identified as its non-key link.	**1**

Notes: Also accept ClubID → AdvisorID → AdvisorName. AdvisorID is not a superkey of the original membership table.

8(b)(iii) Third Normal Form [4]

STUDENT(StudentID, StudentName)
ADVISOR(AdvisorID, AdvisorName)
CLUB(ClubID, ClubName, AdvisorID)
CLUB_MEMBERSHIP(StudentID, ClubID, JoiningDate)

A. STUDENT table with correct underlined key.	**1**
B. ADVISOR table with correct underlined key.	**1**
C. CLUB table with correct underlined key.	**1**
D. CLUB_MEMBERSHIP with correct composite underlined key.	**1**

Notes: Sensible table names and any table order are accepted. Arrows are not requested in this part. JoiningDate belongs to the membership, not STUDENT or CLUB. Do not retain AdvisorName in CLUB.

Total for Question 8: 15 marks

Question 9 - Python programming

Working interpretation: the main key reproduces the printed example: 12 days → Rs.225. See the alternative-credit note on the next page.

9(a) calculateFine function [5]

```
def calculateFine(lateDays):
    if lateDays <= 5:
        return 0
    elif lateDays <= 10:
        return (lateDays - 5) * 10
    else:
        fine = 50 + (lateDays - 5) * 25
        return min(fine, 500)
```

A. Correct function definition, parameter and return mechanism.	**1**
B. 0-5 days produces zero.	**1**
C. 6-10 days produces (lateDays - 5) * 10.	**1**
D. Over 10 days produces 50 + (lateDays - 5) * 25.	**1**
E. Correctly limits the returned fine to 500.	**1**

Notes: Accept equivalent branching and cap logic. The component must operate for the stated input range. Do not require negative-input validation.

9(b) Create fineList [4]

```
fineList = []
for memberNo, lateDays in lateReturns:
    fine = calculateFine(lateDays)
    if fine > 0:
        fineList.append((memberNo, fine))
```

A. Initializes fineList as a list.	**1**
B. Traverses the supplied records and obtains both fields.	**1**
C. Calls calculateFine with the late-day value.	**1**
D. Appends (memberNo, fine) tuples only for positive fines; requires A and a valid traversal.	**1**

Notes: Accept indexing instead of tuple unpacking, or an equivalent list comprehension. Use the function from (a); do not re-penalise a consistently carried-forward fine formula.

Result for the supplied list:

```
[("M003", 50), ("M004", 425), ("M005", 500)]
```

Question 9 (continued)

9(c) Sequential search [4]

```
def searchFine(fineList, memberNo):
    for number, fine in fineList:
        if number == memberNo:
            return fine
    return -1
```

A. Function with the required input parameters.	**1**
B. Sequentially examines the records.	**1**
C. Compares membership number and returns its associated fine; requires B.	**1**
D. Returns -1 only after all records have been exhausted; requires B.	**1**

Notes: A while-loop search is accepted. An early return -1 inside the loop fails D. A dictionary lookup does not meet the sequential-search requirement. Preserve the supplied function name.

9(d) Outputs [2]

```
0
20
300
125
```

A. 0.5 for each correct output, in order.	**2**

Notes: The last output is taken from the supplied search list, not recalculated.

Alternative-credit rule for 9(a), (b) and (d)

A conventional progressive reading gives five free days, five days at Rs.10, then days after day 10 at Rs.25. It does not reproduce the printed 12-day example, but is a plausible response to the conflicting tariff wording. This model accepts it consistently:

```
return min(50 + (lateDays - 10) * 25, 500)
```

Use this expression for the over-10 branch only. Apply the same component maxima. The corresponding fineList and outputs are:

```
[("M003", 50), ("M004", 300), ("M005", 500)]

0
20
175
125
```

Notes: Do not mix the two interpretations within a single answer. A consistent choice of either interpretation is not penalised again in a later part. This alternative-credit rule is proposed for this model; it is not an official ruling.

Total for Question 9: 15 marks

Question 10 - Systems and web applications

10(a)(i) Functional and non-functional requirements [2]

Functional - any one exact statement:
“Students should be able to view available events through the online system.”
“Students should provide their Admission Number, Name, and Email address when registering for an event.”
“The system should check existing registrations before accepting a new registration request.”
“The system should provide confirmation after a successful registration.”
“Administrators should be able to view event registration details and participant counts.”

Non-functional:
“The system should be accessible at any time from different locations.”

A. One correct functional requirement using the supplied wording.	**1**
B. Correct non-functional requirement using the supplied wording.	**1**

Notes: The question expressly requires exact wording. Ignore harmless punctuation differences; do not accept an invented requirement.

10(a)(ii)(1) Waterfall limitation [1]

Late changes require revisiting completed stages, increasing rework, cost or delay.

A. Relates the limitation to changes after coding has begun.	**1**

10(a)(ii)(2) Suitable model and reason [2]

Agile / iterative development: repeated feedback and short increments accommodate changing requirements.

A. Suitable model: Agile / iterative-incremental / evolutionary prototyping.	**1**
B. Explains how it accommodates the requested changes; requires a suitable choice.	**1**

10(b)(i) Contents of $result [1]

The query result set for SELECT * FROM EVENT, represented by a mysqli_result object.

A. Result set / object containing access to the selected event records.	**1**

Notes: “All records returned from EVENT” is sufficient at this level. Do not require the API class name. Not a single fetched row, connection object or Boolean true.

10(b)(ii) Select two columns [2]

```
SELECT EventID, EventName FROM EVENT;
```

A. Correct two-column projection.	**1**
B. Correct source table and complete SELECT syntax.	**1**

Notes: SQL keyword case and final semicolon may be ignored. Attribute names must identify the supplied columns.

Question 10 (continued)

10(b)(iii) Hidden inputs [2]

Carry the student ID and selected event ID to register.php in the submitted form, without requiring visible/repeated entry.

A. Transfers the student and event identifiers with the POST request.	**1**
B. Preserves the registration context without visible/retyped controls.	**1**

Notes: Hidden fields are not encryption or authentication. Do not award the purpose mark for “keeps the data secure from the user”.

10(b)(iv) Insert registration [2]

SQL text to place inside the existing PHP string:

```
INSERT INTO REGISTRATION
    (StudentID, EventID, RegistrationDate)
VALUES ('$studentID', '$eventID', CURRENT_DATE)
```

A. Correct INSERT target and column list, omitting automatic RegistrationID.	**1**
B. Matching values and valid current-date expression with correct syntax.	**1**

Notes: CURDATE() is accepted in MySQL. Accept an equivalent explicit column order with values in the corresponding order. Do not require the candidate to rewrite the entire PHP application.

10(c)(i) Direct versus parallel deployment [1]

A. Direct generally costs less; parallel costs more through duplicate operation.	**0.5**
B. Direct has higher operational risk; parallel offers a working fallback and comparison.	**0.5**

Notes: Each credited comparison must include both methods, explicitly or unambiguously.

10(c)(ii) Recommendation for the bank [2]

No. Failure could interrupt critical banking transactions/services without the concurrently operating old system as fallback.

A. Does not recommend direct deployment for the stated critical banking replacement.	**1**
B. Relevant operational-risk explanation; requires A.	**1**

Notes: A reasoned conditional recommendation may earn 2 if it explicitly establishes tested rollback/recovery and explains why immediate cutover is acceptable. “It is cheaper” alone is not sufficient justification for this scenario.

Total for Question 10: 15 marks

Final check

Part A: 40 marks. Part B: best four questions, 60 marks. Maximum Paper II score: 100. Retain half marks until the final total.