

Class: XI Session: 2026-27
SAMPLE QUESTION PAPER
Informatics Practices (065)
Marking Scheme

Time allowed: 3 Hours
Maximum Marks: 70

General Instructions:

- The paper is divided into 4 Sections- A, B, C, D and E.
- Section A, consists of 21 questions (1 to 21). Each question carries 1 Mark.
- Section B, consists of 7 questions (22 to 28). Each question carries 2 Marks.
- Section C, consists of 4 questions (29 to 32). Each question carries 3 Marks.
- Section D, consists of 2 questions (33 to 34). Each question carries 4 Marks.
- Section E, consists of 3 questions (35 to 37). Each question carries 5 Marks.
- All programming questions are to be answered using Python Language only.

Q.NO.	QUESTION	MARKS
1	b) Only the file's address is marked as free, while the actual data may still exist.	1
2	c) Adding a new column	1
3	b) Utility Software	1
4	d) str	1
5	a) bit < nibble < byte < word	1
6	c) BETWEEN	1
7	a) 13	1
8	a) NULL	1
9	b) [10, 20, 30, [40, 50]]	1
10	c) The change is rejected because the other computers still contain the original ledger.	1
11	b) Only if both conditions are true.	1
12	a) Touch Screen	1
13	a) Show tables	1
14	b) A set of valid values for an attribute	1

15	a) NLP	1
16	b) CHAR	1
17	c) It must be unique and cannot be NULL for any row	1
18	c) Every row within that targeted attribute field is updated.	1
19	c) SaaS (Software as a Service)	1
20	c) A is True but R is False	1
21	b) Both A and R are true and R is not the correct explanation for A	1

Section B		
Q.NO.	QUESTION	MARKS
22	a) Remaining Available Space = 200 MB Approx (500 MB - 300 MB), i.e. 204,800 KB b) True	1+1=2
23	Advantages of DBMS (any 4): Reduced Data Redundancy, Data Consistency, Data Sharing, Data Security, Easy Maintenance & Backups, Enforcement of Standards (½ mark for each, max 2 marks)	2
24	a) Grid Computing (or distributed computing) b) 1. Veracity 2. Variety	1 ½+½
25	Output: @ N * @ A (½ mark for each correct line, cap at 2 marks)	2
26	import numpy as np runs = np.array([220, 180, 250, 300, 275, 190]) print(runs[-2]) # Output: 275 OR Output: [40 60] [15 15 15]	1+1

27	(i) ALTER TABLE CAR ADD FuelType VARCHAR(20); (ii) DDL (Data Definition Language) command OR (i) Yes, a table can have multiple Alternate Keys (e.g. in CAR table, if CarID is Primary Key, RegistrationNo and EngineNo are Alternate Keys) (ii) DROP TABLE CAR;	1+1
28	a) 1. Correct (chained assignment supported) 2. Incorrect - raises TypeError b) 1. 512 2. DAVDAVDAV OR Corrected code: temps = [32, 35, 29, 40, 31] average_temp = sum(temps) / len(temps) print("The average temperature is:", average_temp) (½ mark for each correct line)	½+½ ½+½
SECTION C		
29	a) ['Green', 'Blue', 'Cyan'] b) ['Magenta', 'Cyan', 'Blue', 'Green', 'Red'] c) ['Blue', 'Cyan', 'Magenta']	1+1+1
30	a) for count in range(1, 6, 2): print(count) b) Output: 0 6 OR Password validation program: magic_word = "open sesame" user_input = "" while user_input != magic_word: user_input = input("Speak the correct magic word to open: ") print("The treasure chest opens!")	2+1
31	a) Bonus: 1500, NULL, 5000, NULL, 8000 b) DELETE FROM EmployeeData WHERE Emp_ID = 501 OR Emp_ID = 502; c) SELECT Emp_ID FROM EmployeeData WHERE Salary IS NULL; OR a) ALTER TABLE teacher ADD PRIMARY KEY (TID); b) 1. CREATE DATABASE SportsArena 2. DESCRIBE Sports;	1+1+1
32	a) CREATE TABLE Sports (SportID INT PRIMARY KEY, SportName VARCHAR(30), Budget DECIMAL(10,2), MatchDate DATE); b) Error: Table already exists (a database cannot contain two tables with the same name)	2+1

SECTION D

33	a) Specific Purpose (Customized) Software - integrates specific campus infrastructure requirements not available in ready-made software b) Operating Systems (System Software) c) Word Processor / Text Editor (e.g. MS Word, Writer, Google Docs)	2(1+1) 1 1
34	Starting with runs = [30, 15, 90, 45, 60] a) runs.sort() → [15, 30, 45, 60, 90] b) runs.pop() → [15, 30, 45, 60] c) runs.insert(1, 75) → [15, 75, 30, 45, 60] d) print(max(runs) + sum(runs)) → 300 (1 mark for each correct answer)	4x1=4

SECTION E

35	a) Degree: 5 Cardinality: 7 b) SELECT * FROM ENTERTAINMENT WHERE NOT Genre = 'Crime'; c) DELETE FROM ENTERTAINMENT WHERE Price < 299; d) ALTER TABLE ENTERTAINMENT DROP PRICE; e) UPDATE ENTERTAINMENT SET Price = Price + Price*10/100;	5x1=5
----	--	-------

36

- a) Primary Key: BookID
 b) Yes - if two books share Author and Title, neither can serve as a candidate key (duplicate values violate uniqueness)
 c) INSERT INTO book VALUES ('B105','Machine Learning',NULL,475);
 OR
 INSERT INTO book (BookID,Title,Price) VALUES ('B105','Machine Learning',475);
 d) SELECT Title, Price-(Price*10/100) FROM Book;
 e) 1 or True
 (1 mark for each correct answer)
 OR

Output of alternative MySQL queries based on table CINEMA:

a) Output:

MOVIE NAME	GENRE
Sky Force	Action
Housefull 5	Comedy

b)

MOVIE NAME	NEW PRICE
Chhava	300
<u>Stree 2</u>	270
Sky Force	350
Sitaare Zameen Par	230
Raid 2	330
Housefull 5	290

c)

MOVIE NAME	GENRE
Chhava	Historical
Sky Force	Action
Raid 2	Crime Thriller

d)

MOVIE NAME	TICKETPRICE * SEATS
Chhava	30750
<u>Stree 2</u>	17600
Sky Force	45000
Sitaare Zameen Par	17100
Raid 2	30800
Housefull 5	26400

5x1=5

	e) SELECT Movie_name, TicketPrice FROM CINEMA WHERE TicketPrice >= 250;	
37	Dictionary operations: a) student["Club"] = "Dramatics" b) student["age"] = 18 c) print(student.get("Name")) → None (case-sensitive key) d) del student['marks']; print(student) → dictionary without 'marks' key 5. e) student.update({"name": "Amita", "house no": 123, "sector": 49, "pincode": 121001}) (1 mark for each correct answer) OR List operations: a) "Data Science" in BookTitle b) len(BookTitle) c) BookTitle.index("Cyber Safety") d) BookTitle.remove("SQL Guide") e) BookTitle.sort(reverse=True) (1 mark for each correct answer)	5x1=5