

CS23304- JAVA PROGRAMMING

LAB TEST (III SEMESTER - Q BATCH)

Date: 3-9-2025

Marks: 25

Draw the following table in your first page

MARK SPLIT-UP

Q.NO.	COMMENTS	MARK
Total		

Answer All Questions

Q1. Design a system that processes a **2D array of encrypted strings**. Each string contains **alphanumeric characters** and is supposed to follow a strict pattern. You must:

1. **Validate the format** of each string. (3)
2. **Decrypt** valid strings using a custom decryption algorithm.(5)
3. **Handle** all exceptions gracefully without halting the program.(5)
4. At the end, return a new 2D array of **decrypted strings**, preserving the matrix structure. (2)

Requirements:

Input: A 2D array of strings: String[][] encryptedMatrix.

String Format Rule:

Each string must follow the pattern: 3 uppercase letters followed by 3 digits (e.g., "ABC123").

Decryption Rule:

For valid strings:

- Shift each letter **backward by 1** (e.g., 'B' → 'A', 'A' → 'Z')
- Invert each digit (e.g., '0' → '9', '1' → '8', ..., '9' → '0')

Example:

Encrypted: "BCD345" → Decrypted: "ABC654"

Custom Exceptions:

Define and use the following exceptions:

- `InvalidFormatException` – when a string doesn't match the required pattern.
- `DecryptionException` – for any error during decryption.
- `MatrixProcessingException` – wraps all other exceptions during matrix processing.

Sample Input

```
{
    {"ABC123", "XYZ789", "INVALID1"},
    {null, "DEF456", "GHI000"},
    {"JKL999", "MNO111", "123ABC"}
}
```

Sample Output

```
{
    {"ZAB876", "WXY210", null},
    {null, "CDE543", "FGH999"},
    {"IJK000", "LMN888", null}
}
```

Q2. Design a **smart vehicle simulation system** using Java **inheritance and polymorphism**, where different types of smart vehicles (e.g., electric cars, drones, and trucks) can perform actions such as moving, charging, and reporting status.

Requirements

Abstract Class: SmartVehicle

Represents the base for all smart vehicles.

Properties:

String id

double batteryLevel

Methods:

void charge(double amount)

abstract void move(double distance)

abstract String getStatus()

Subclasses (Each must override move() and getStatus()):

a. ElectricCar

Property: int passengerCount

Battery Consumption: distance \times 0.5

getStatus(): Returns battery level and passenger count

b. AutonomousDrone

Property: double altitude

Battery Consumption: distance \times 1.0 + altitude \times 0.2

getStatus(): Returns battery level and current altitude

c. SelfDrivingTruck

Property: double cargoWeight

Battery Consumption: distance \times (0.7 + cargoWeight \times 0.1)

getStatus(): Returns battery level and cargo weight

Write a **main class** where you:

Create an array or list of SmartVehicle objects.

(2)

Initialize each object with different data.

(1)

Invoke move() and getStatus() for each object using a **polymorphic reference**

(e.g., SmartVehicle vehicle = new ElectricCar(...);)

(5)

Ensure output is dynamically bound at runtime.

(2)