**PG-DBDA ( Re- Exam Aug 2024 Batch)**

**Date : 5/02/2025**

**Time : 4:30 PM to 6 PM**

**Section : 20 Marks**

### **Problem Statement: Bank Account Management System**

**Objective:**Develop a **BankAccount** class that simulates **basic banking operations** including **deposit, withdrawal, and balance inquiry** while implementing **exception handling** for invalid transactions.

### **Problem Description:**

A bank wants to develop a **Bank Account Management System** that allows customers to **create an account, deposit money, withdraw money**, and **view their balance**. The system should ensure that:

1. **A new account must have a non-negative initial balance.**
2. **Deposits must be positive amounts.**
3. **Withdrawals must not exceed the account balance.**
4. **Exception handling must be implemented** to prevent errors such as:
   * Withdrawing more money than available
   * Depositing or withdrawing a negative amount

### **Requirements:**

1. **Create a class BankAccount** with the following private data members:
   * int acc no (Account Number)
   * double balance (Account Balance)
2. **Constructors:**
   * Initialize the account with an **account number** and **starting balance**.
   * If the **initial balance is negative**, throw an exception.
3. **Functions to Implement:**
   * deposit(double amount): Adds money to the balance (must be positive).
   * withdraw(double amount): Deducts money from the balance (if sufficient funds exist).
   * show(): Displays the **account number and current balance**.
4. **Exception Handling:**
   * If the user tries to **withdraw more than the balance**, an **exception should be raised**.
   * If the user **enters a negative amount for deposit/withdrawal**, an **exception should be raised**.
5. **Main Program (main()):**
   * Create a BankAccount object.
   * Perform **deposit and withdrawal transactions**.
   * Demonstrate **exception handling** when an invalid transaction occurs.

**Section B: 20 Marks**

### **Problem Statement: Shape Rendering System Using Object-Oriented Programming (OOP)**

**A graphics software company wants to develop a shape rendering system that can dynamically handle different geometric shapes using Object-Oriented Programming (OOP) principles. The system should allow drawing and erasing of various shapes while maintaining a structured and scalable design.**

## **📝 System Requirements:**

### **1️. Base Class (Shape)**

* The system should contain a base class Shape that acts as a parent class for all geometric shapes.
* **It should contain two methods:**
  + draw(): Displays a message indicating that the shape is being drawn.
  + erase(): Displays a message indicating that the shape is being erased.

### **2. Subclasses (Circle, Triangle, Square)**

* The system should include three specific shapes (Circle, Triangle, Square).
* Each subclass should define its own version of draw() and erase() methods to provide unique behavior for each shape.

### **3️. Object-Oriented Design Implementation**

* The program should be designed in a way that new shapes can be added easily in the future without modifying the existing structure.
* It should ensure that the Shape class provides a common interface for all specific shapes.

### **4️. Implementation in the main() Method**

* The program should allow creating and handling multiple shapes dynamically.
* Users should be able to create objects for different shapes and call their respective draw() and erase() methods.