1. **Problem Statement: Develop a BankAccount class that implements core banking operations: ○ balanceEnquiry(): Displays the current account balance. ○ withdraw(): Deducts the specified amount from the account balance. ○ deposit(): Adds the specified amount to the account balance.**

**Program:**

import java.util.Scanner;

class LowBalanceException extends Exception {

    public LowBalanceException(String message) {

        super(message);

    }

}

class NegativeNumberException extends Exception {

    public NegativeNumberException(String message) {

        super(message);

    }

}

class BankAccount {

    private double balance;

    public BankAccount(double initialBalance) {

        this.balance = initialBalance;

    }

    public void balanceEnquiry() {

        System.out.println("Current Balance: ₹" + balance);

    }

    public void withdraw(double amount) throws LowBalanceException, NegativeNumberException {

        if (amount < 0) {

            throw new NegativeNumberException("Cannot withdraw a negative amount!");

        }

        if (amount > balance) {

            throw new LowBalanceException("Insufficient balance for withdrawal!");

        }

        balance -= amount;

        System.out.println("$" + amount + " withdrawn successfully.");

    }

    public void deposit(double amount) throws NegativeNumberException {

        if (amount < 0) {

            throw new NegativeNumberException("Cannot deposit a negative amount!");

        }

        balance += amount;

        System.out.println("$" + amount + " deposited successfully.");

    }

}

public class BankAccountDemo {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        BankAccount account = new BankAccount(5000); // Starting with ₹5000

        while (true) {

            System.out.println("\n--- Bank Menu ---");

            System.out.println("1. Balance Enquiry");

            System.out.println("2. Deposit");

            System.out.println("3. Withdraw");

            System.out.println("4. Exit");

            System.out.print("Choose an option: ");

            int choice = sc.nextInt();

            try {

                switch (choice) {

                    case 1:

                        account.balanceEnquiry();

                        break;

                    case 2:

                        System.out.print("Enter amount to deposit: ");

                        double depositAmount = sc.nextDouble();

                        account.deposit(depositAmount);

                        break;

                    case 3:

                        System.out.print("Enter amount to withdraw: ");

                        double withdrawAmount = sc.nextDouble();

                        account.withdraw(withdrawAmount);

                        break;

                    case 4:

                        System.out.println("Thank you for banking with us!");

                        sc.close();

                        return;

                    default:

                        System.out.println("Invalid option. Try again.");

                }

            } catch (LowBalanceException | NegativeNumberException e) {

                System.out.println("Error: " + e.getMessage());

            }

        }

    }

}

**Output:**

--- Bank Menu ---

1. Balance Enquiry

2. Deposit

3. Withdraw

4. Exit

Choose an option: 2

Enter amount to deposit: 1000

$1000.0 deposited successfully.

--- Bank Menu ---

1. Balance Enquiry

2. Deposit

3. Withdraw

4. Exit

Choose an option: 1

Current Balance: $6000.0

--- Bank Menu ---

1. Balance Enquiry

2. Deposit

3. Withdraw

4. Exit

Choose an option:

1. **Write a Java program with a method that takes an integer as input. If the number is odd, the method should throw a custom exception (OddNumberException). Handle this exception in the main**

**program :**

import java.util.Scanner;

class OddNumberException extends Exception {

    public OddNumberException(String message) {

        super(message);

    }

}

public class OddNumberDemo {

    static void checkEven(int number) throws OddNumberException {

        if (number % 2 != 0) {

            throw new OddNumberException("Entered number is odd!");

        } else {

            System.out.println("Entered number is even.");

        }

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter an integer: ");

        int num = sc.nextInt();

        try {

            checkEven(num);

        } catch (OddNumberException e) {

            System.out.println("Exception Caught: " + e.getMessage());

        } finally {

            sc.close();

        }

    }

}

**Output:**

Enter an integer: 11

Exception Caught: Entered number is odd!

1. **Create a package ExceptionHandlingDemo containing classes Calculator and DivisionException. ○ The Calculator class should have a method divide(int a, int b) that performs division. ○ If b is zero, throw a custom exception DivisionException with an appropriate error message. ○ Handle the exception in the main program and display an error message instead of crashing.**

package ExceptionHandlingDemo;

public class CalculatorDemo {

public int divide(int a, int b) throws DivisionException {

if (b == 0) {

throw new DivisionException("Cannot divide by zero!");

}

return a / b;

}

}

package ExceptionHandlingDemo;

public class DivisionException extends Exception {

    public DivisionException(String message) {

        super(message);

    }

}

Main:

import ExceptionHandlingDemo.CalculatorDemo;

import ExceptionHandlingDemo.DivisionException;

import java.util.Scanner;

public class MainCalculatorDemo {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        CalculatorDemo calc = new CalculatorDemo();

        System.out.print("Enter numerator: ");

        int a = sc.nextInt();

        System.out.print("Enter denominator: ");

        int b = sc.nextInt();

        try {

            int result = calc.divide(a, b);

            System.out.println("Result: " + result);

        } catch (DivisionException e) {

            System.out.println("Error: " + e.getMessage());

        } finally {

            sc.close();

        }

    }

}

**Output:**

Enter numerator: 12

Enter denominator: 0

Error: Cannot divide by zero!