Experiment - 3

Student Name: Ayush Ranjan **UID:** 23BCS10187
Branch: BE-CSE **Section/Group:** KRG-2-B
Semester: 5th **Date of Performance:** 9/9/25
Subject Name: PBLJ **Subject Code:** 23CSH-304

1. **Aim:** Write a Java program to simulate an ATM withdrawal system. The program should:
   a) Ask the user to enter their PIN.
   b) Allow withdrawal if the PIN is correct and the balance is sufficient.
   c) Throw exceptions for invalid PIN or insufficient balance.
   d) Ensure the system always shows the remaining balance, even if an exception occurs.

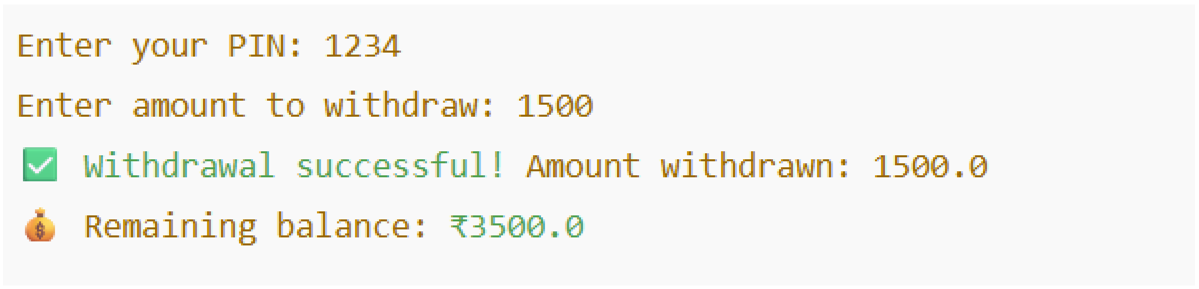
2. **Objective:** Implement nested try-catch blocks and create meaningful exception messages.

3. **Procedure:**
   a) Prompt the user to enter their ATM PIN.
   b) Check if the PIN is correct.
   c) If valid, prompt for withdrawal amount.
   d) Check whether the withdrawal amount is less than or equal to the balance.
   e) If not, throw a custom Insufficient Balance Exception.
   f) Use finally to print the current balance irrespective of the exception.

4. **Code -**

```
import java.util.Scanner;

class InvalidPinException extends Exception {
    public InvalidPinException(String message) {
        super(message);
    }
}

class InsufficientBalanceException extends Exception {
    public InsufficientBalanceException(String message) {
        super(message);
    }
}

public class ATMSystem {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int correctPin = 1234;
        double balance = 5000.0;
        try {
            System.out.print("Enter your PIN: ");
            int enteredPin = sc.nextInt();


            if (enteredPin != correctPin) {
                throw new InvalidPinException("❌ Invalid PIN! Please try again.");
            }

            System.out.print("Enter amount to withdraw: ");
            double amount = sc.nextDouble();

            if (amount > balance) {
                throw new InsufficientBalanceException("❌ Insufficient balance!");
            }

            balance -= amount;
            System.out.println("✅ Withdrawal successful! Amount withdrawn: " + amount);

        } catch (InvalidPinException | InsufficientBalanceException e) {
            System.out.println(e.getMessage());
        } finally {
            System.out.println("💰 Remaining balance: ₹" + balance);
        }

        sc.close();
    }
}
```

5. Output -



```
Enter your PIN: 1234
Enter amount to withdraw: 1500
✅ Withdrawal successful! Amount withdrawn: 1500.0
💰 Remaining balance: ₹3500.0
```

6. Learning Outcomes:

a) Gained understanding of how to use exception handling (try-catch-finally) in real-world applications.

b) Learnt to create and use custom exception classes for specific error conditions.

c) Gained practical knowledge of user input handling using Scanner.

d) Learnt to apply conditional logic to validate PINs and check account balance.

e) Understood the importance of the finally block to execute essential code, such as displaying the remaining balance, even after exceptions.