TASK 3

INTERNSHIP ON JAVA PROGRAMMING

Q) 1.Create a class to represent the ATM machine.

2. Design the user interface for the ATM, including options such as withdrawing, depositing, and checking the balance.

3. Implement methods for each option, such as withdraw(amount), deposit(amount), and checkBalance().

4. Create a class to represent the user's bank account, which stores the account balance.

5. Connect the ATM class with the user's bank account class to access and modify the account balance.

6. Validate user input to ensure it is within acceptable limits (e.g., sufficient balance for withdrawals).

7. Display appropriate messages to the user based on their chosen options and the success or failure of their transactions.

PROGRAM:

```java
import java.util.Scanner;


// BankAccount class representing user's bank account
class BankAccount {
   private double balance;


   // Constructor to initialize balance
   public BankAccount(double balance) {
      this.balance = balance;
   }


   // Method to deposit money into the account
   public void deposit(double amount) {
      balance += amount;
   }
```

```java
    // Method to withdraw money from the account
    public boolean withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            return true;
        } else {
            return false;
        }
    }

    // Method to check balance
    public double checkBalance() {
        return balance;
    }
}

// ATM class representing the ATM machine
public class ATM {
    private BankAccount account;

    // Constructor to initialize account
    public ATM(BankAccount account) {
        this.account = account;
    }

    // Method to display ATM options
    public void displayOptions() {
        System.out.println("ATM Options:");
        System.out.println("1. Withdraw");
```

```java
        System.out.println("2. Deposit");
        System.out.println("3. Check Balance");
        System.out.println("4. Exit");
    }


    // Method to withdraw money
    public void withdraw(double amount) {
        if (account.withdraw(amount)) {
            System.out.println("Withdrawal successful. Remaining balance: " +
account.checkBalance());
        } else {
            System.out.println("Insufficient balance. Withdrawal failed.");
        }
    }


    // Method to deposit money
    public void deposit(double amount) {
        account.deposit(amount);
        System.out.println("Deposit successful. Current balance: " +
account.checkBalance());
    }


    // Method to check balance
    public void checkBalance() {
        System.out.println("Current balance: " + account.checkBalance());
    }


    // Main method to run the ATM interface
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
```

```java
// Create a bank account with initial balance
BankAccount userAccount = new BankAccount(1000);

// Create ATM object
ATM atm = new ATM(userAccount);

boolean exit = false;

while (!exit) {
    atm.displayOptions();
    System.out.print("Enter your choice: ");
    int choice = scanner.nextInt();

    switch (choice) {
        case 1:
            System.out.print("Enter amount to withdraw: ");
            double withdrawAmount = scanner.nextDouble();
            atm.withdraw(withdrawAmount);
            break;
        case 2:
            System.out.print("Enter amount to deposit: ");
            double depositAmount = scanner.nextDouble();
            atm.deposit(depositAmount);
            break;
        case 3:
            atm.checkBalance();
            break;
        case 4:
            exit = true;
            System.out.println("Thank you for using the ATM. Goodbye!");
```

```java
                break;
            default:
                System.out.println("Invalid choice. Please enter a valid option.");
        }
    }


    scanner.close();
    }
}
```

OUTPUT:

ATM Options:

1. Withdraw

2. Deposit

3. Check Balance

4. Exit

Enter your choice: 1

Enter amount to withdraw: 200

Withdrawal successful. Remaining balance: 800.0

ATM Options:

1. Withdraw

2. Deposit

3. Check Balance

4. Exit

Enter your choice: 2

Enter amount to deposit: 500

Deposit successful. Current balance: 1300.0

ATM Options:

1. Withdraw

2. Deposit

3. Check Balance

4. Exit

Enter your choice: 3

Current balance: 1300.0

ATM Options:

1. Withdraw

2. Deposit

3. Check Balance

4. Exit

Enter your choice: 4

Thank you for using the ATM. Goodbye!

=== Code Execution Successful ===

The above program creates an ATM interface with options to withdraw, deposit, check balance, and exit. It connects the ATM class with the BankAccount class to access and modify the user's account balance. User input is validated, and appropriate messages are displayed based on the chosen options and the success or failure of transactions.