Week 8 - S8 - Core OOP - Abstract Class And Interface - Practice Problem

**Name:**Ramesh Harisabapathi Chettiar

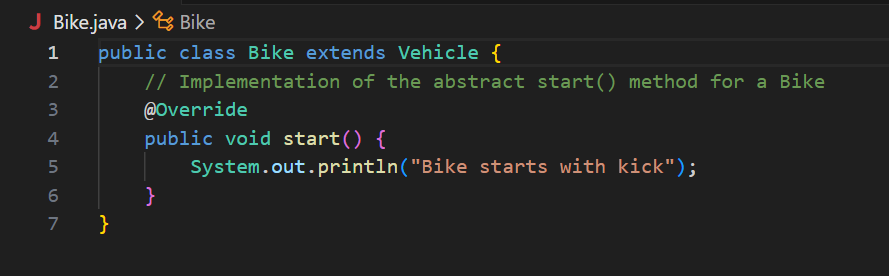
**Date of Submission:**06/10/2025

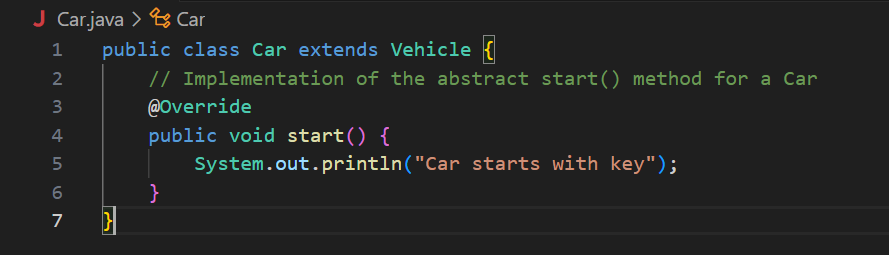
QNO1🡪

Create an abstract class Vehicle with an abstract method start(). Subclasses Car

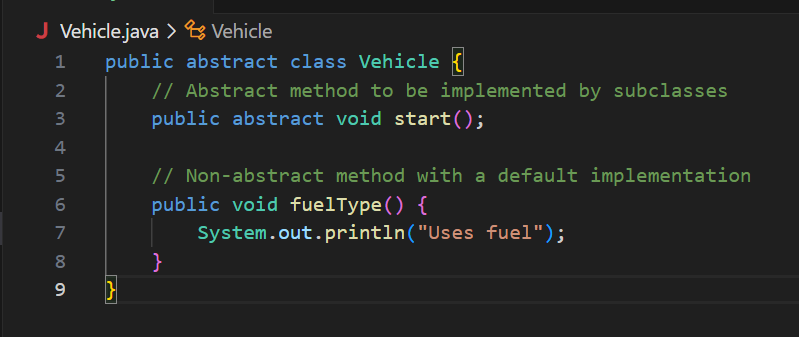
and Bike will extend Vehicle and provide their own implementations for start().

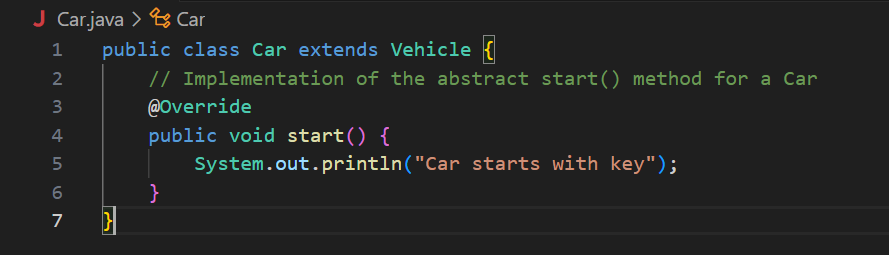
Demonstrate abstraction by using Vehicle references to call the methods.

Bike.java

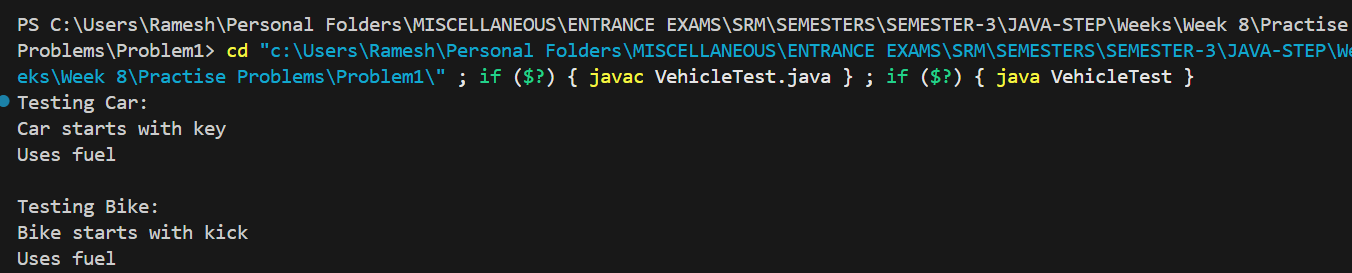
Car.java

Vehicle.java



VehicleTest.java

**OUTPUT🡪**

****

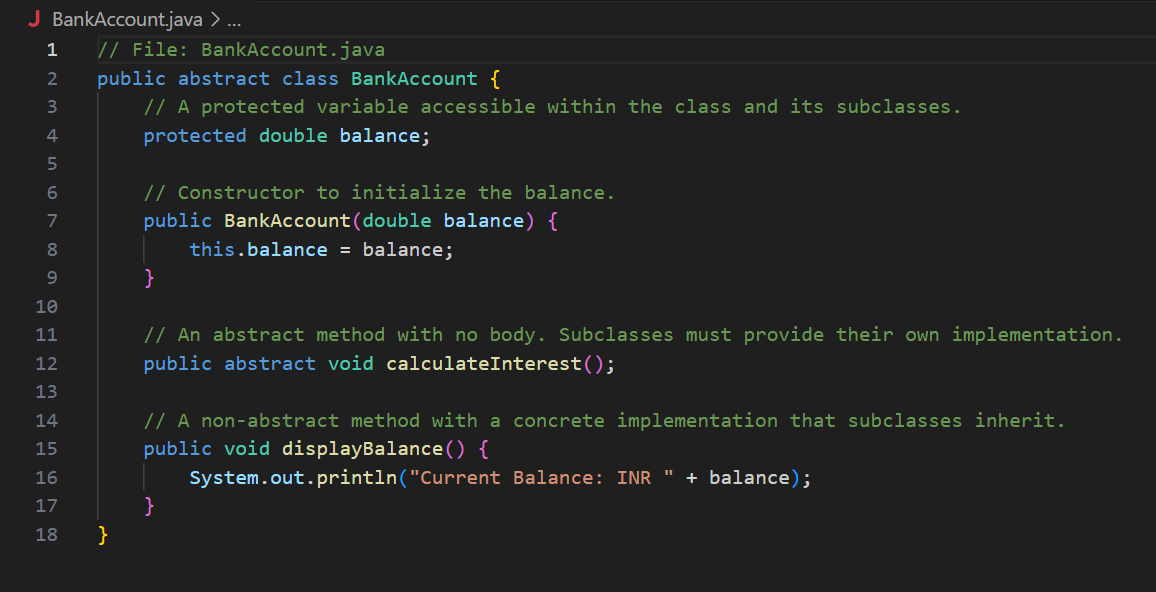
**QNO2🡪**

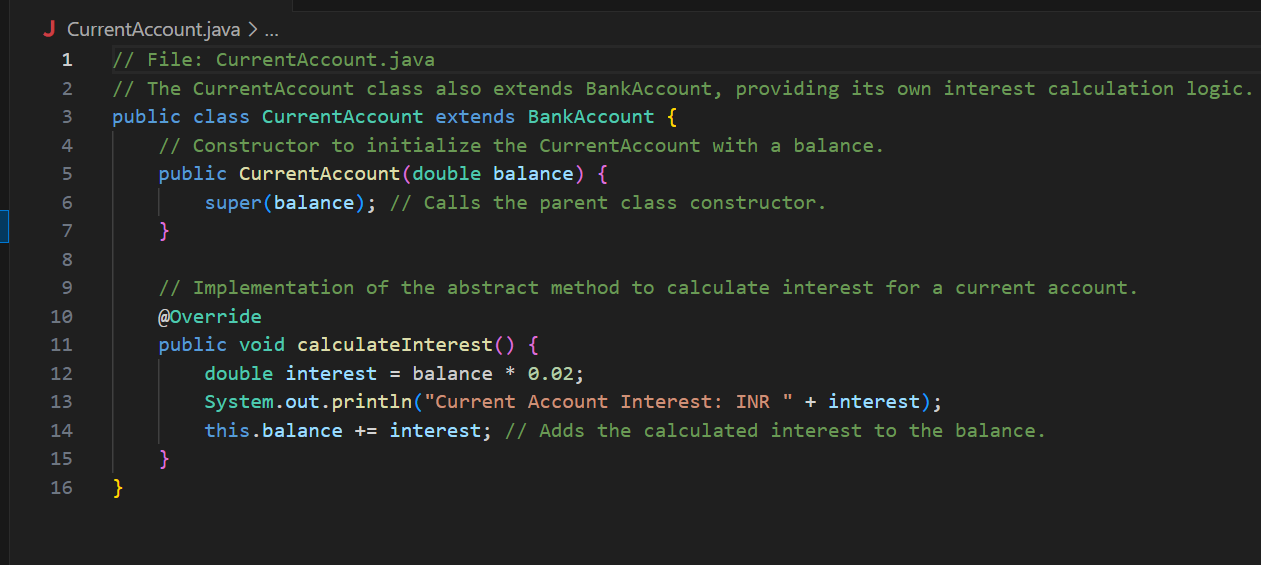
Design an abstract class BankAccount with abstract method

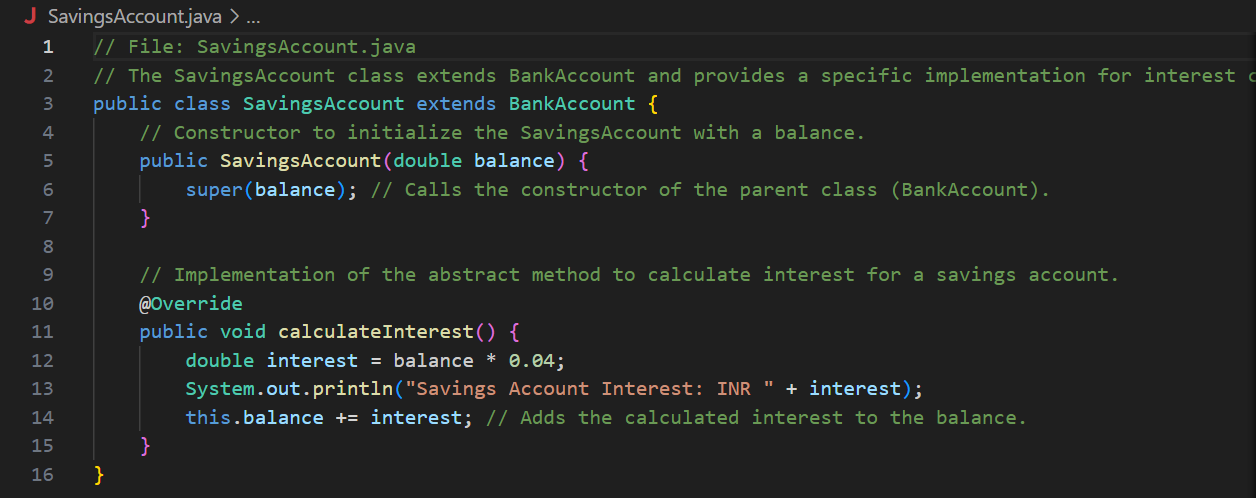
calculateInterest(). Subclasses SavingsAccount and CurrentAccount

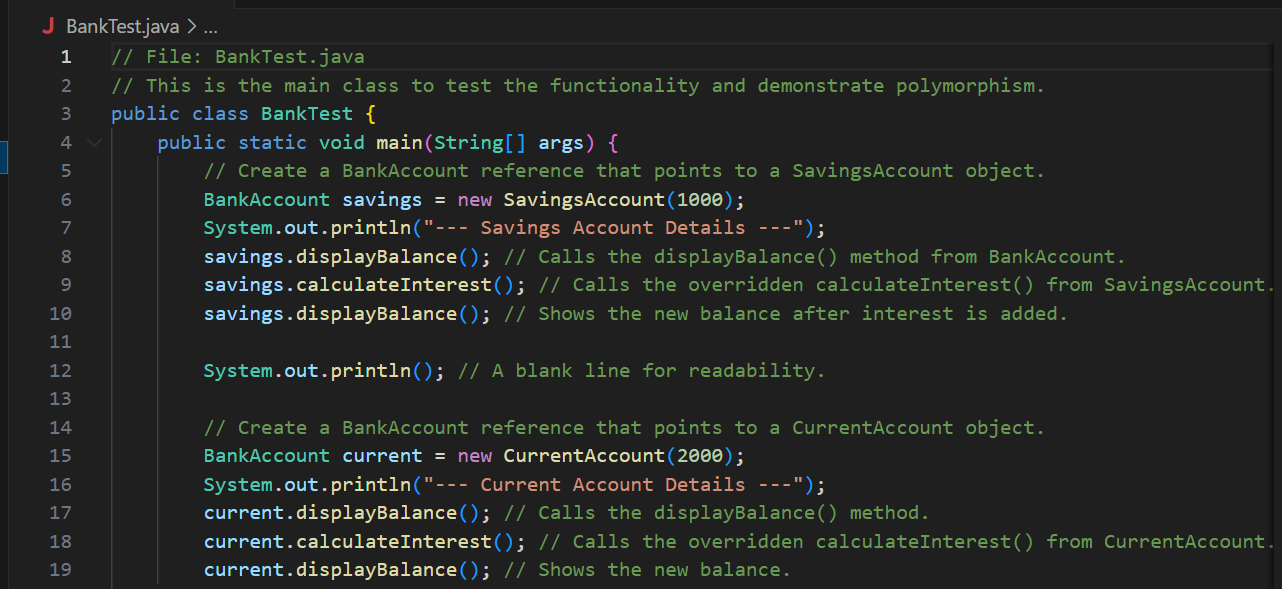
should implement it differently. Demonstrate abstraction by handling different account

types.

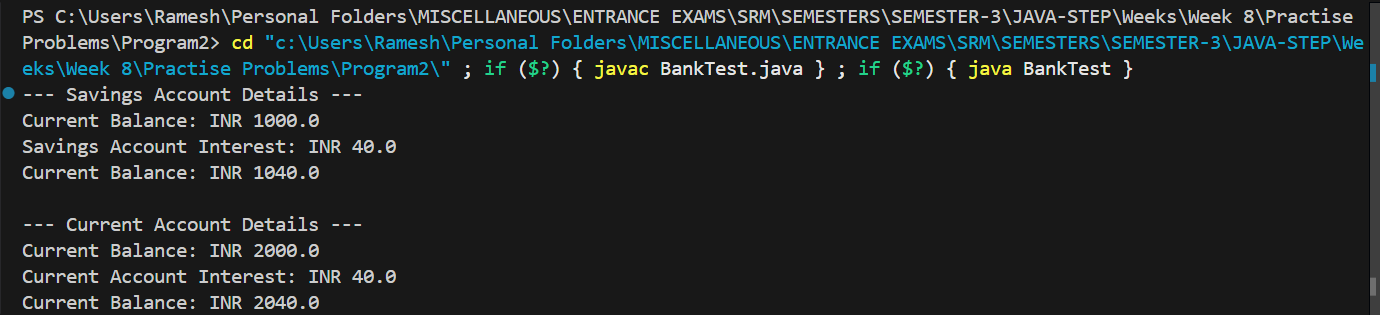
BankAccount.java

CurrentAccount.java

SavingsAccount.java

BankTest.java

**OUTPUT🡪**

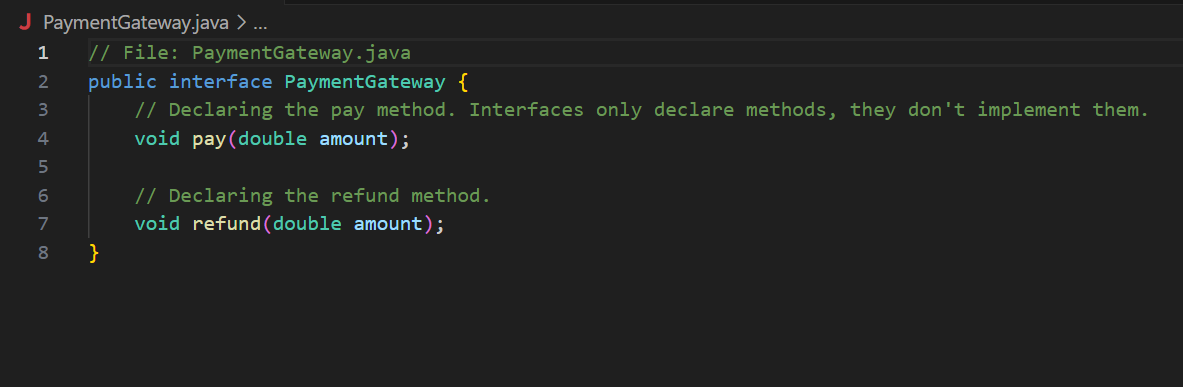
****

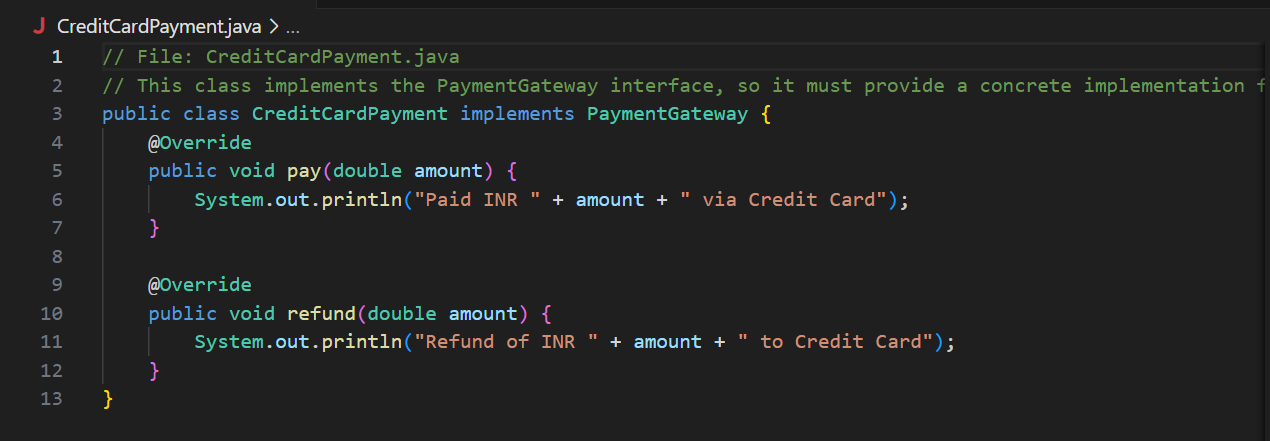
**QNO3🡪**

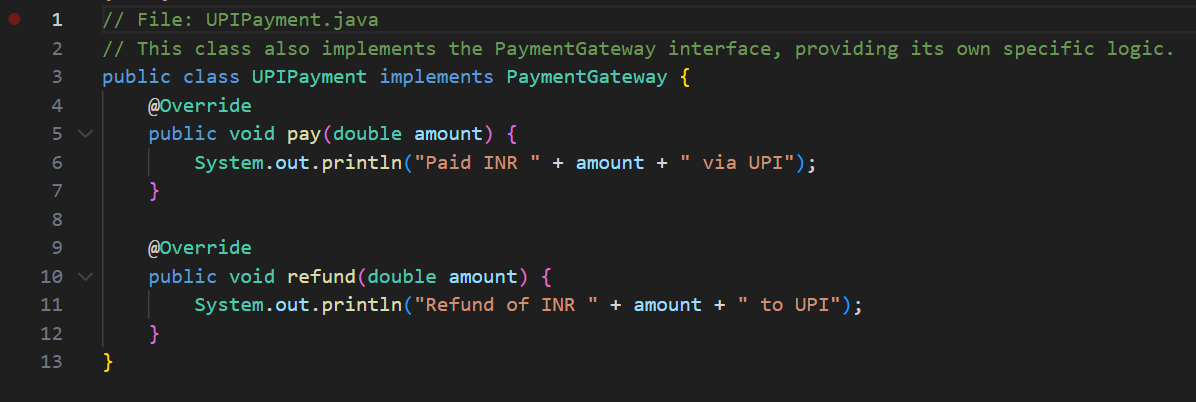
**Create an interface PaymentGateway with methods pay() and refund().**

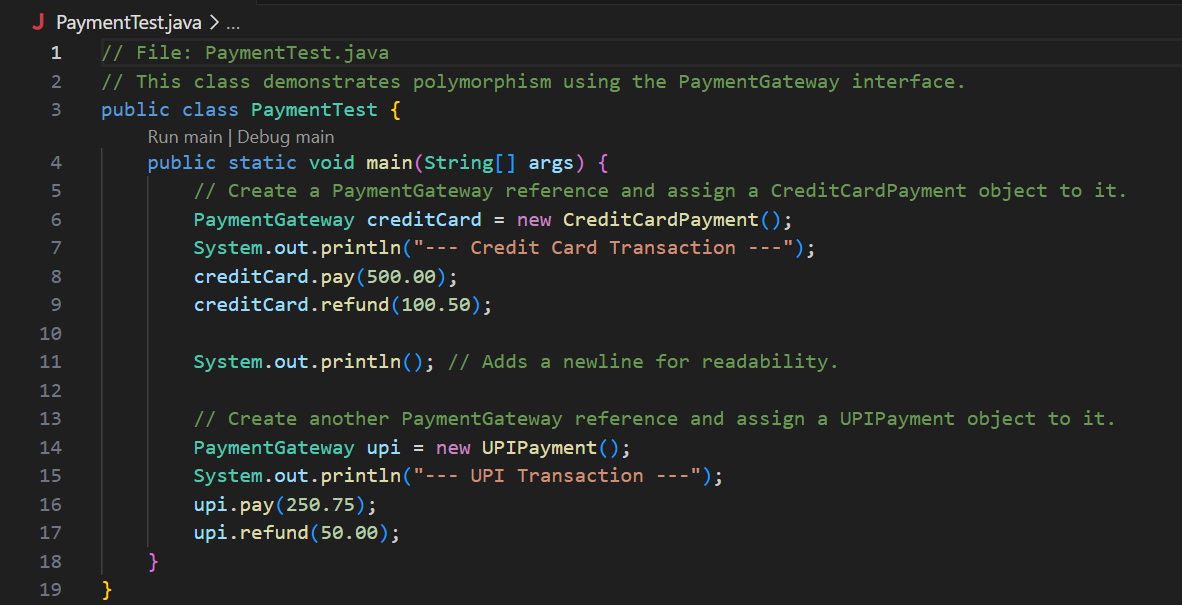
**Implement this interface in CreditCardPayment and UPIPayment. Demonstrate**

**multiple payment methods using interfaces.**

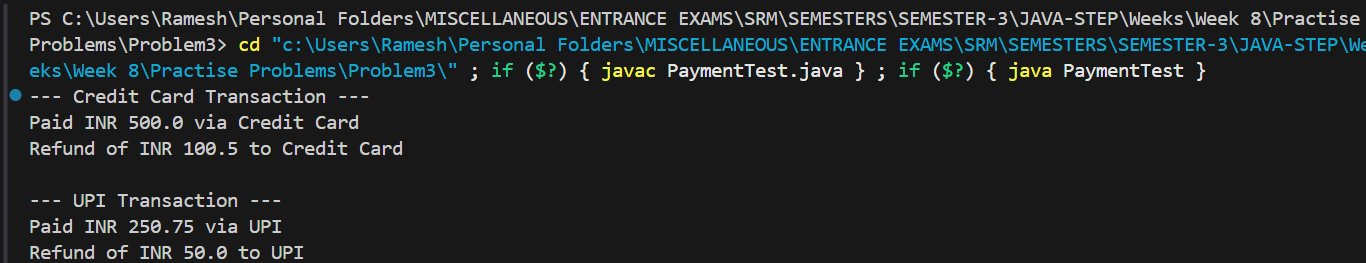
**PaymentGateway.java**

**CreditCardPayment.java**

**UPIPayment.java**

**PaymentTest.java**

**OUTPUT🡪**

****