**LabSession**

**Hitarth Patel**

**150096724046**

**Jensen Huang**

1.Write a Java program to create a class called "BankAccount" with instance variables 'accountNumber' and balance, and static variables 'bankName' and 'interestRate'. Provide static methods to get and set the static variables. Create several 'BankAccount' objects and print their details along with the static variables.

→

import java.util.Scanner;

public class BankAccount {

private int accountNumber;

private double balance;

private static String bankName = "XYZ Bank";

private static double interestRate = 0.05;

public BankAccount(int *accountNumber*, double *balance*) {

this.accountNumber = *accountNumber*;

this.balance = *balance*;

}

public static String **getBankName**() {

return bankName;

}

public static double **getInterestRate**() {

return interestRate;

}

public void **printDetails**() {

System.out.println("Account Number: " + accountNumber);

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

System.out.println("Bank Name: " + bankName);

System.out.println("Interest Rate: " + interestRate);

}

public static void **main**(String[] *args*) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter account number for account 1: ");

int accountNumber1 = scanner.nextInt();

System.out.println("Enter balance for account 1: ");

double balance1 = scanner.nextDouble();

BankAccount account1 = new BankAccount(accountNumber1, balance1);

System.out.println("Enter account number for account 2: ");

int accountNumber2 = scanner.nextInt();

System.out.println("Enter balance for account 2: ");

double balance2 = scanner.nextDouble();

BankAccount account2 = new BankAccount(accountNumber2, balance2);

System.out.println("Enter account number for account 3: ");

int accountNumber3 = scanner.nextInt();

System.out.println("Enter balance for account 3: ");

double balance3 = scanner.nextDouble();

BankAccount account3 = new BankAccount(accountNumber3, balance3);

System.out.println("Bank Details:");

System.out.println("Bank Name: " + BankAccount.getBankName());

System.out.println("Interest Rate: " + BankAccount.getInterestRate());

System.out.println("\nAccount 1 Details:");

account1.printDetails();

System.out.println("\nAccount 2 Details:");

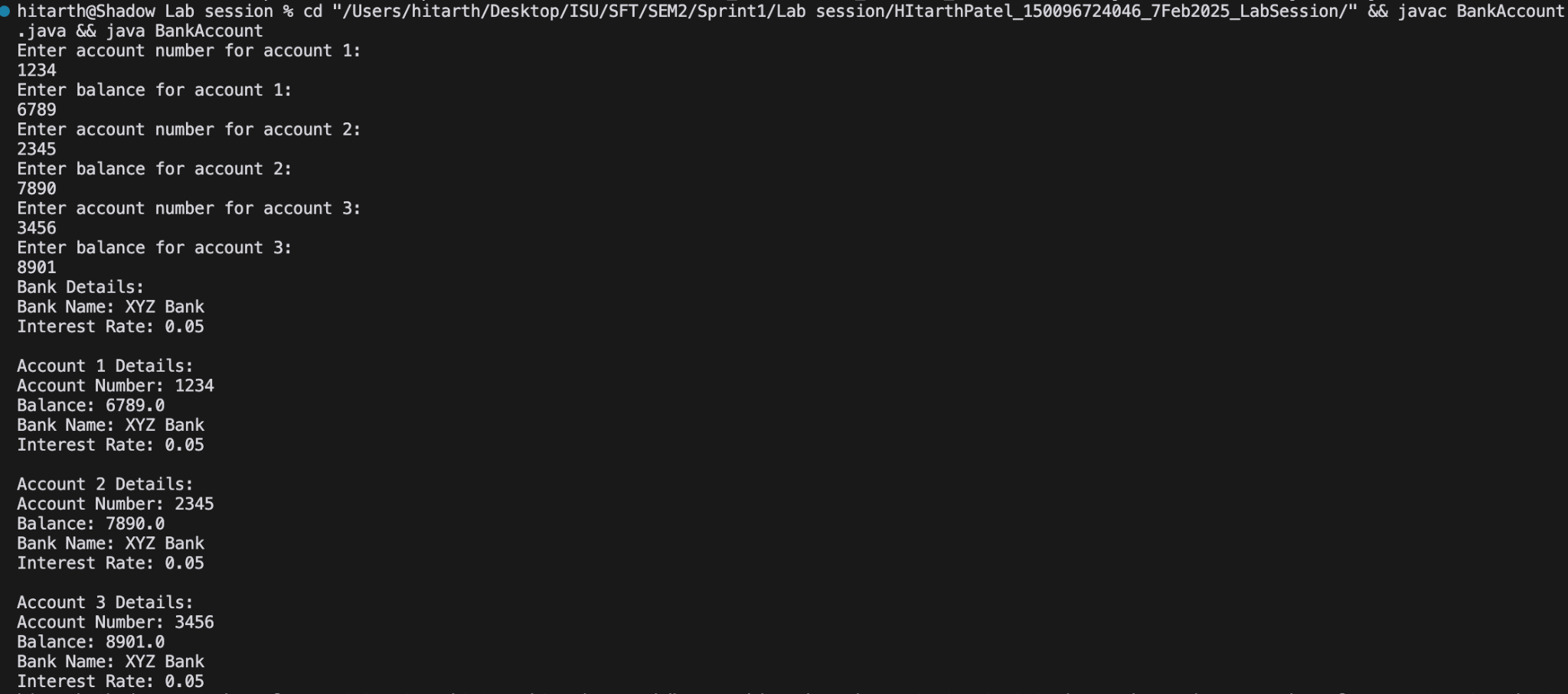
account2.printDetails();

System.out.println("\nAccount 3 Details:");

account3.printDetails();

scanner.close();

}

}

2.Write a Java program to create a vehicle class hierarchy. The base class should be Vehicle, with subclasses Truck, Car and Motorcycle. Each subclass should have properties such as make, model, year, and fuel type. Implement methods for calculating fuel efficiency, distance traveled, and maximum speed.

→

import java.util.Scanner;

class Vehicle {

protected String make;

protected String model;

protected int year;

protected String fuelType;

public Vehicle(String *make*, String *model*, int *year*, String *fuelType*) {

this.make = *make*;

this.model = *model*;

this.year = *year*;

this.fuelType = *fuelType*;

}

public double **calculateFuelEfficiency**() {

return 0;

}

public double **calculateDistanceTraveled**() {

return 0;

}

public double **calculateMaximumSpeed**() {

return 0;

}

}

class Truck extends Vehicle {

private double fuelCapacity;

private double fuelEfficiency;

public Truck(String *make*, String *model*, int *year*, String *fuelType*, double *fuelCapacity*, double *fuelEfficiency*) {

super(*make*, *model*, *year*, *fuelType*);

this.fuelCapacity = *fuelCapacity*;

this.fuelEfficiency = *fuelEfficiency*;

}

public double **calculateFuelEfficiency**() {

return fuelEfficiency;

}

public double **calculateDistanceTraveled**() {

return fuelCapacity \* fuelEfficiency;

}

public double **calculateMaximumSpeed**() {

return 100;

}

}

class Car extends Vehicle {

private double fuelCapacity;

private double fuelEfficiency;

public Car(String *make*, String *model*, int *year*, String *fuelType*, double *fuelCapacity*, double *fuelEfficiency*) {

super(*make*, *model*, *year*, *fuelType*);

this.fuelCapacity = *fuelCapacity*;

this.fuelEfficiency = *fuelEfficiency*;

}

public double **calculateFuelEfficiency**() {

return fuelEfficiency;

}

public double **calculateDistanceTraveled**() {

return fuelCapacity \* fuelEfficiency;

}

public double **calculateMaximumSpeed**() {

return 120;

}

}

class Motorcycle extends Vehicle {

private double fuelCapacity;

private double fuelEfficiency;

public Motorcycle(String *make*, String *model*, int *year*, String *fuelType*, double *fuelCapacity*, double *fuelEfficiency*) {

super(*make*, *model*, *year*, *fuelType*);

this.fuelCapacity = *fuelCapacity*;

this.fuelEfficiency = *fuelEfficiency*;

}

public double **calculateFuelEfficiency**() {

return fuelEfficiency;

}

public double **calculateDistanceTraveled**() {

return fuelCapacity \* fuelEfficiency;

}

public double **calculateMaximumSpeed**() {

return 150;

}

}

public class Cars {

public static void **main**(String[] *args*) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter vehicle type (Truck, Car, Motorcycle): ");

String vehicleType = scanner.nextLine();

System.out.println("Enter vehicle make: ");

String make = scanner.nextLine();

System.out.println("Enter vehicle model: ");

String model = scanner.nextLine();

System.out.println("Enter vehicle year: ");

int year = scanner.nextInt();

scanner.nextLine();

System.out.println("Enter vehicle fuel type: ");

String fuelType = scanner.nextLine();

System.out.println("Enter vehicle fuel capacity: ");

double fuelCapacity = scanner.nextDouble();

scanner.nextLine();

System.out.println("Enter vehicle fuel efficiency: ");

double fuelEfficiency = scanner.nextDouble();

scanner.nextLine();

Vehicle vehicle = null;

if (vehicleType.equalsIgnoreCase("Truck")) {

vehicle = new Truck(make, model, year, fuelType, fuelCapacity, fuelEfficiency);

} else if (vehicleType.equalsIgnoreCase("Car")) {

vehicle = new Car(make, model, year, fuelType, fuelCapacity, fuelEfficiency);

} else if (vehicleType.equalsIgnoreCase("Motorcycle")) {

vehicle = new Motorcycle(make, model, year, fuelType, fuelCapacity, fuelEfficiency);

}

if (vehicle != null) {

System.out.println("Vehicle Details:");

System.out.println("Make: " + vehicle.make);

System.out.println("Model: " + vehicle.model);

System.out.println("Year: " + vehicle.year);

System.out.println("Fuel Type: " + vehicle.fuelType);

System.out.println("Fuel Efficiency: " + vehicle.calculateFuelEfficiency());

System.out.println("Distance Traveled: " + vehicle.calculateDistanceTraveled());

System.out.println("Maximum Speed: " + vehicle.calculateMaximumSpeed());

} else {

System.out.println("Invalid vehicle type.");

}

scanner.close();

}

}