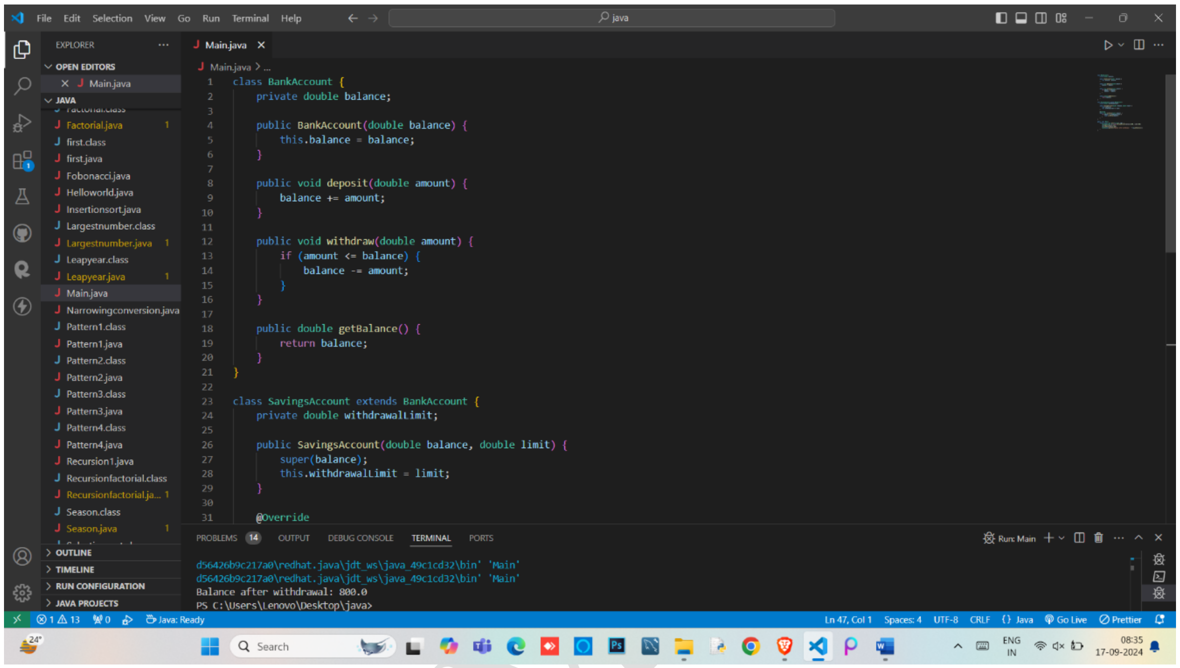
Assignment No- 5

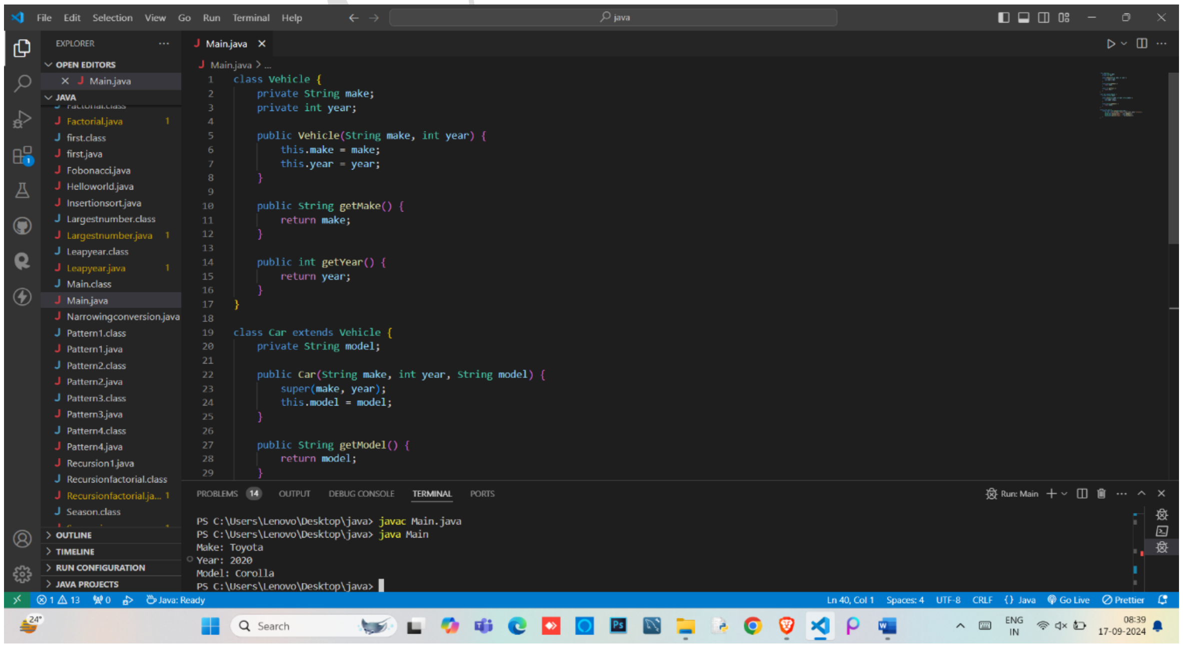
1) Create a base class BankAccount with methods like deposit() and withdraw(). Derive a class SavingsAccount that overrides the withdraw() method to impose a limit on the withdrawal amount. Write a program that demonstrates the use of overridden methods and proper access modifiers & return the details.



```java
class BankAccount {
    private double balance;

    public BankAccount(double balance) {
        this.balance = balance;
    }

    public void deposit(double amount) {
        balance += amount;
    }

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
        }
    }

    public double getBalance() {
        return balance;
    }
}

class SavingsAccount extends BankAccount {
    private double withdrawalLimit;

    public SavingsAccount(double balance, double limit) {
        super(balance);
        this.withdrawalLimit = limit;
    }

    @Override
```

```
d56426b9c217a0\redhat.java\jdt_ws\java_49c1cd32\bin' 'Main'
d56426b9c217a0\redhat.java\jdt_ws\java_49c1cd32\bin' 'Main'
Balance after withdrawal: 800.0
PS C:\Users\Lenovo\Desktop\java>
```

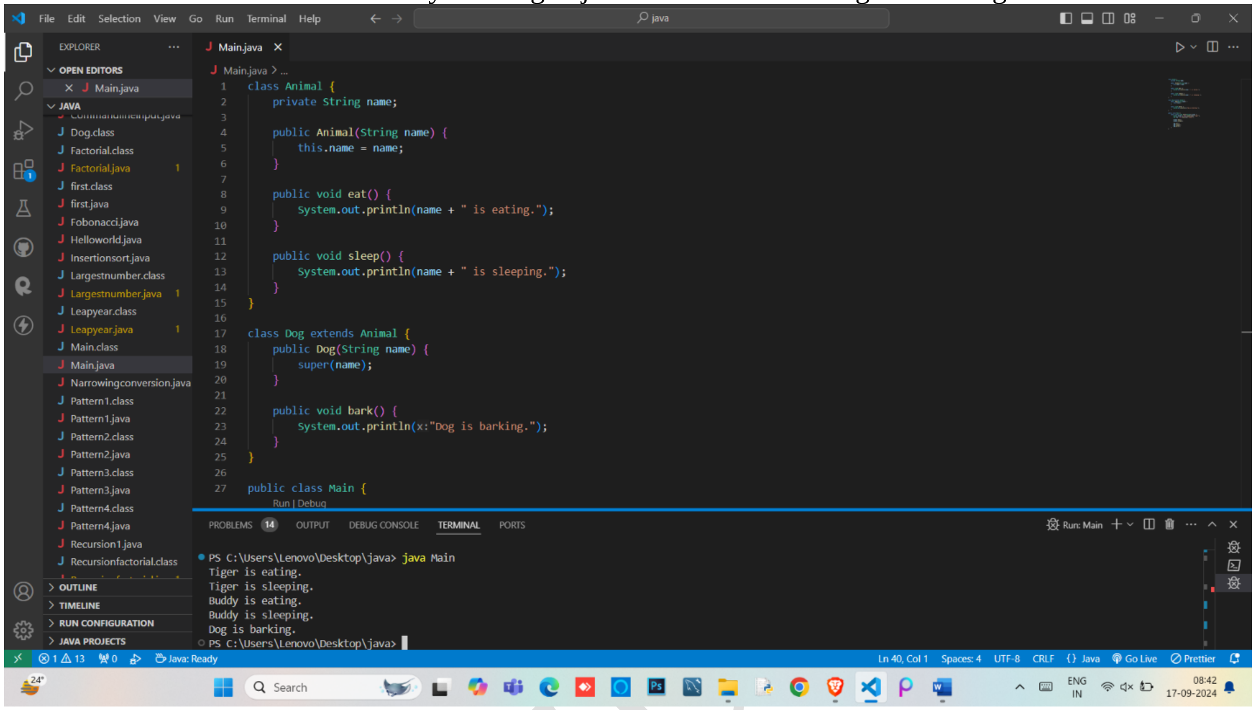
2) Create a base class Vehicle with attributes like make and year. Provide a constructor in Vehicle to initialize these attributes. Derive a class Car that has an additional attribute model and write a constructor that initializes make, year, and model. Write a program to create a Car object and display its details.



```java
class Vehicle {
    private String make;
    private int year;

    public Vehicle(String make, int year) {
        this.make = make;
        this.year = year;
    }

    public String getMake() {
        return make;
    }

    public int getYear() {
        return year;
    }
}

class Car extends Vehicle {
    private String model;

    public Car(String make, int year, String model) {
        super(make, year);
        this.model = model;
    }

    public String getModel() {
        return model;
    }
```

```
PS C:\Users\Lenovo\Desktop\java> javac Main.java
PS C:\Users\Lenovo\Desktop\java> java Main
Make: Toyota
Year: 2020
Model: Corolla
PS C:\Users\Lenovo\Desktop\java>
```

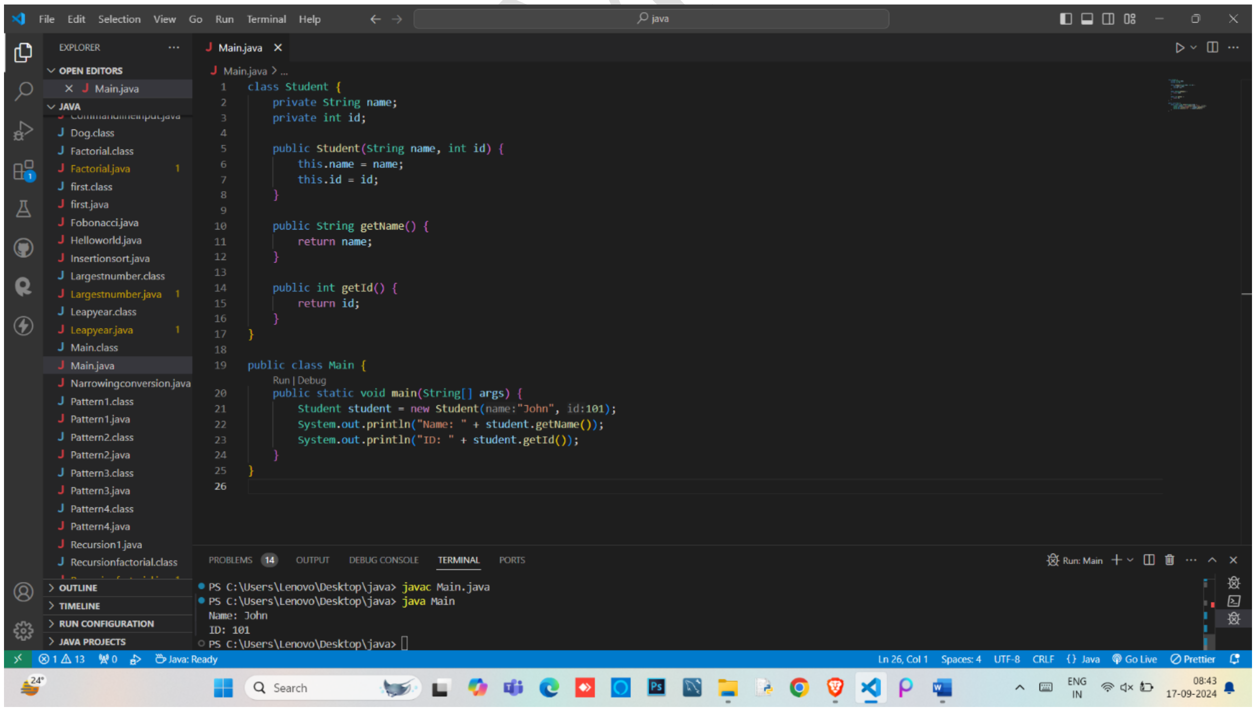
3) Create a base class Animal with attributes like name, and methods like eat() and sleep(). Create a subclass Dog that inherits from Animal and has an additional method bark(). Write a program to demonstrate the use of inheritance by creating objects of Animal and Dog and calling their methods.



```java
class Animal {
    private String name;

    public Animal(String name) {
        this.name = name;
    }

    public void eat() {
        System.out.println(name + " is eating.");
    }

    public void sleep() {
        System.out.println(name + " is sleeping.");
    }
}

class Dog extends Animal {
    public Dog(String name) {
        super(name);
    }

    public void bark() {
        System.out.println(x:"Dog is barking.");
    }
}

public class Main {
```

```
PS C:\Users\Lenovo\Desktop\java> java Main
Tiger is eating.
Tiger is sleeping.
Buddy is eating.
Buddy is sleeping.
Dog is barking.
PS C:\Users\Lenovo\Desktop\java>
```

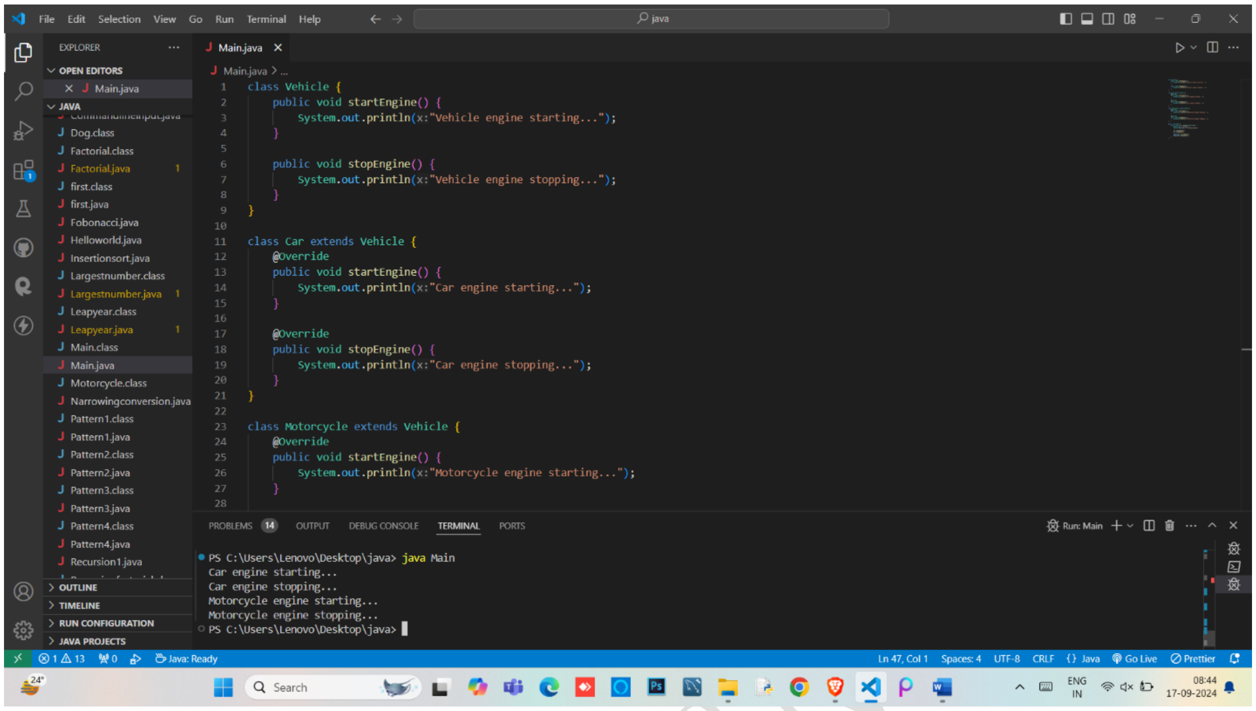
4) Build a class Student which contains details about the Student and compile and run its instance.



```java
class Student {
    private String name;
    private int id;

    public Student(String name, int id) {
        this.name = name;
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public int getId() {
        return id;
    }
}

public class Main {
    public static void main(String[] args) {
        Student student = new Student(name:"John", id:101);
        System.out.println("Name: " + student.getName());
        System.out.println("ID: " + student.getId());
    }
}
```

```
PS C:\Users\Lenovo\Desktop\java> javac Main.java
PS C:\Users\Lenovo\Desktop\java> java Main
Name: John
ID: 101
PS C:\Users\Lenovo\Desktop\java>
```

5) Write a Java program to create a base class Vehicle with methods startEngine() and stopEngine(). Create two subclasses Car and Motorcycle. Override the startEngine() and stopEngine() methods in each subclass to start and stop the engines differently.



```java
class Vehicle {
    public void startEngine() {
        System.out.println(x:"Vehicle engine starting...");
    }

    public void stopEngine() {
        System.out.println(x:"Vehicle engine stopping...");
    }
}

class Car extends Vehicle {
    @Override
    public void startEngine() {
        System.out.println(x:"Car engine starting...");
    }

    @Override
    public void stopEngine() {
        System.out.println(x:"Car engine stopping...");
    }
}

class Motorcycle extends Vehicle {
    @Override
    public void startEngine() {
        System.out.println(x:"Motorcycle engine starting...");
    }

```

```
PS C:\Users\Lenovo\Desktop\java> java Main
Car engine starting...
Car engine stopping...
Motorcycle engine starting...
Motorcycle engine stopping...
PS C:\Users\Lenovo\Desktop\java>
```