OOPS IN JAVA

TUTORIAL - 7

Karthik Krishnan

Roll No: 45

S3 CSE B

Inheritance in Java

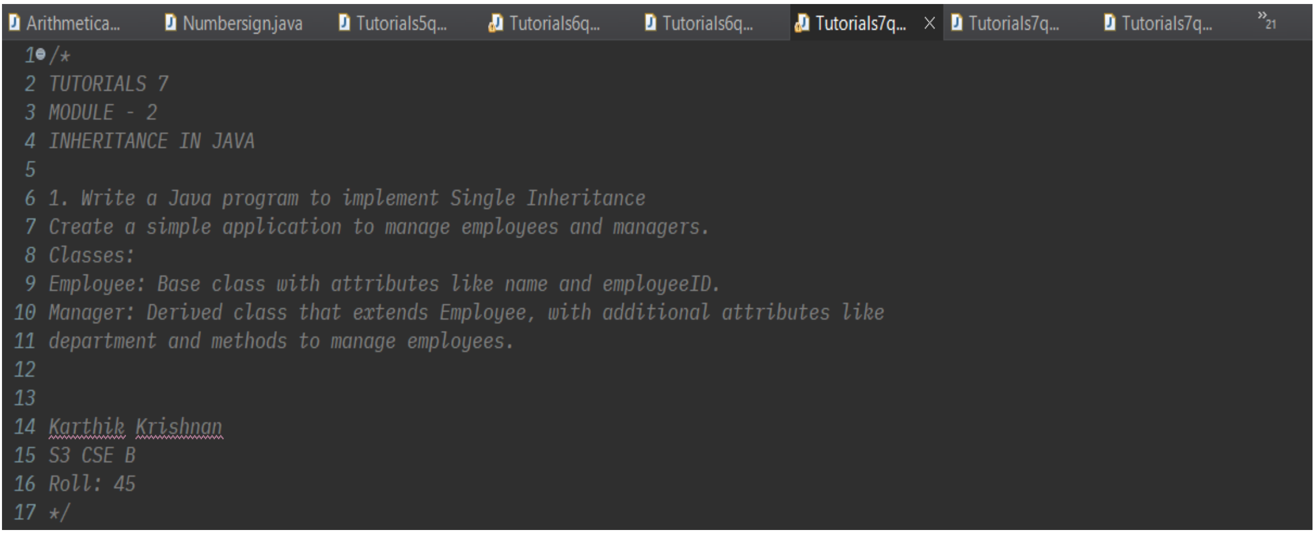
Module - 2

Qn 1) Write a Java program to implement Single Inheritance Create a simple application to manage employees and managers.

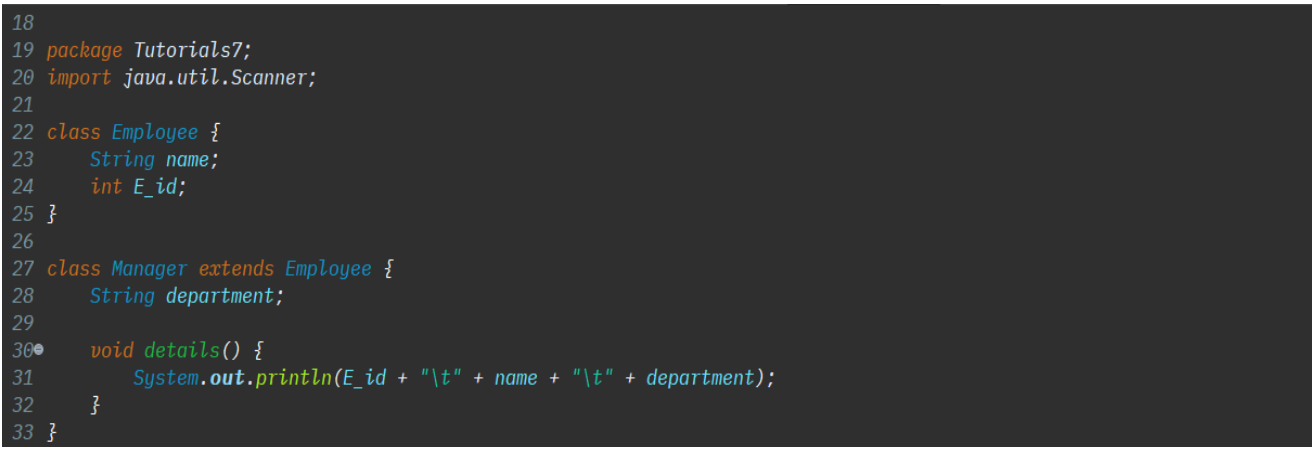
Classes:

Employee: Base class with attributes like name and employeeID.

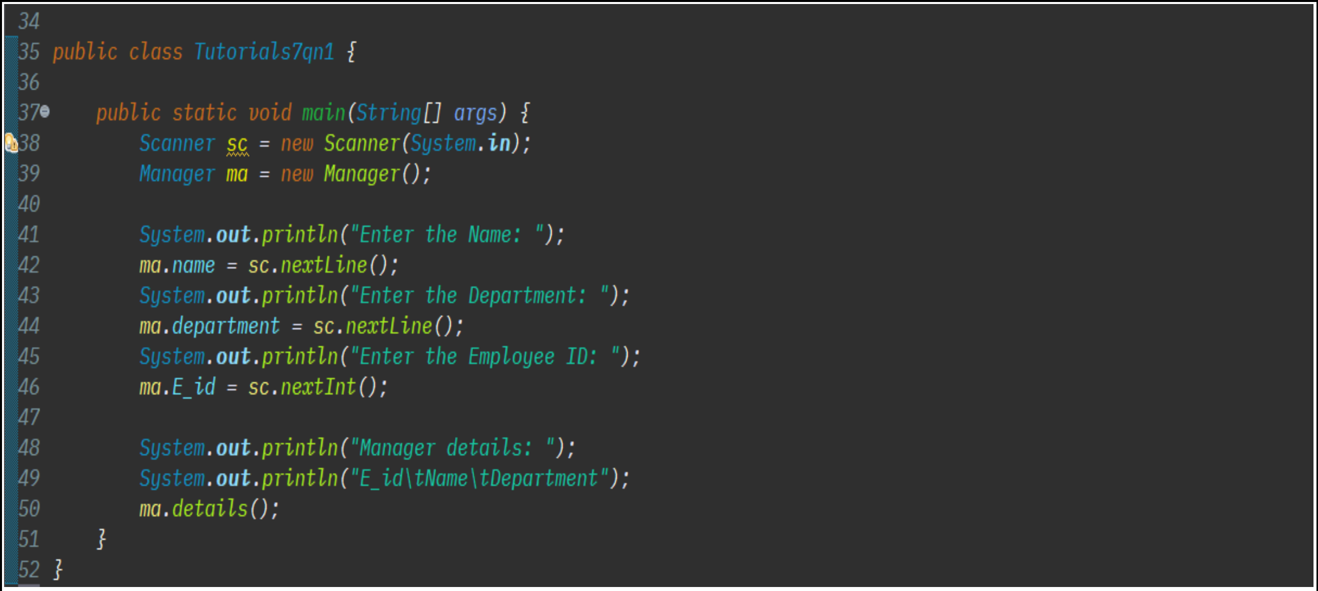
Manager: Derived class that extends Employee, with additional attributes like department and methods to manage employees.



```
/*
TUTORIALS 7
MODULE - 2
INHERITANCE IN JAVA

1. Write a Java program to implement Single Inheritance
Create a simple application to manage employees and managers.
Classes:
Employee: Base class with attributes like name and employeeID.
Manager: Derived class that extends Employee, with additional attributes like
department and methods to manage employees.


Karthik Krishnan
S3 CSE B
Roll: 45
*/
```

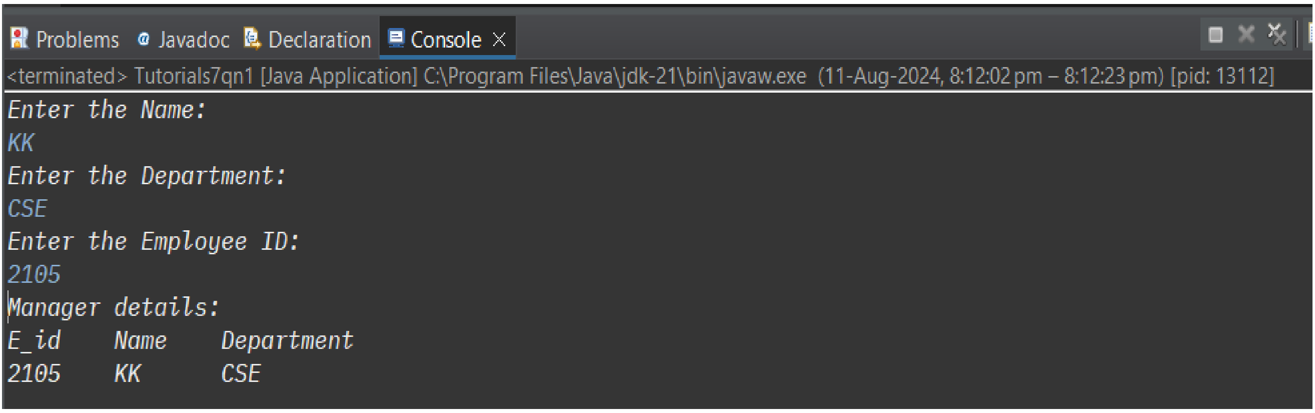


```

package Tutorials7;
import java.util.Scanner;

class Employee {
    String name;
    int E_id;
}

class Manager extends Employee {
    String department;

    void details() {
        System.out.println(E_id + "\t" + name + "\t" + department);
    }
}
```



```java

public class Tutorials7qn1 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Manager ma = new Manager();

        System.out.println("Enter the Name: ");
        ma.name = sc.nextLine();
        System.out.println("Enter the Department: ");
        ma.department = sc.nextLine();
        System.out.println("Enter the Employee ID: ");
        ma.E_id = sc.nextInt();

        System.out.println("Manager details: ");
        System.out.println("E_id\tName\tDepartment");
        ma.details();
    }
}
```

OUTPUT:



<terminated> Tutorials7qn1 [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (11-Aug-2024, 8:12:02 pm – 8:12:23 pm) [pid: 13112]

```
Enter the Name:
KK
Enter the Department:
CSE
Enter the Employee ID:
2105
Manager details:
E_id    Name    Department
2105    KK      CSE
```

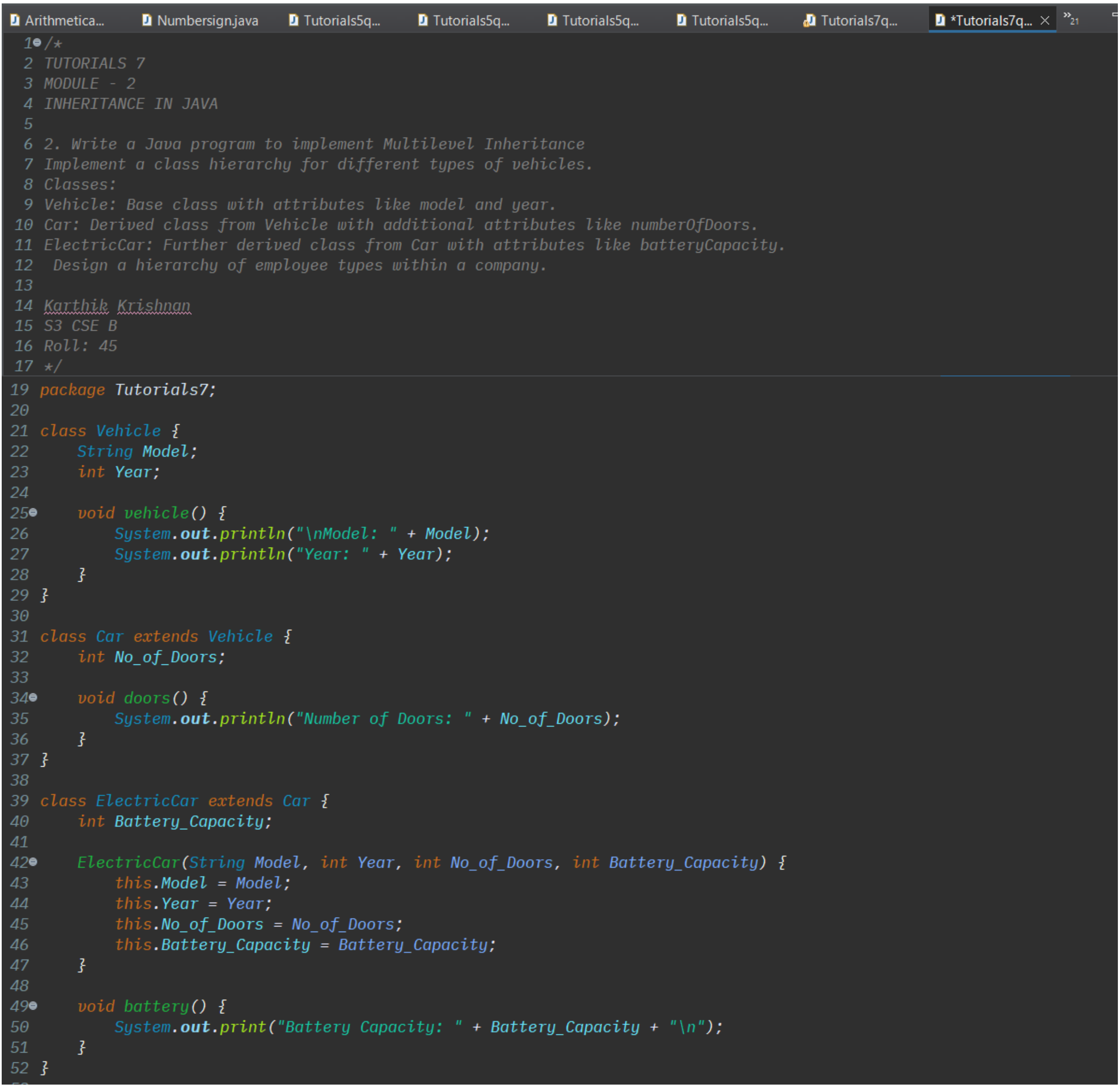
Qn 2) Write a Java program to implement Multilevel Inheritance Implement a class hierarchy for different types of vehicles.

Classes:

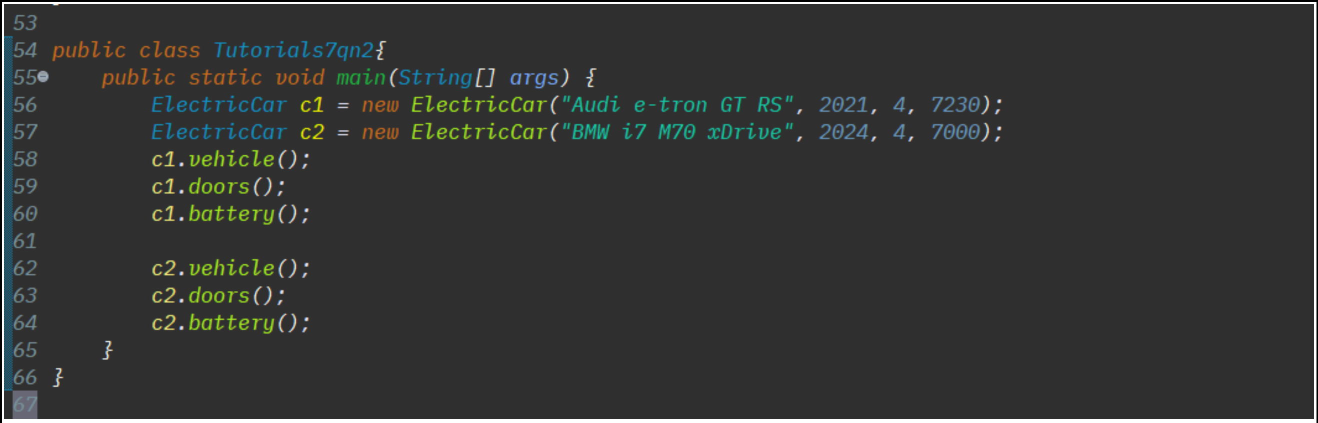
Vehicle: Base class with attributes like model and year.

Car: Derived class from Vehicle with additional attributes like numberOfDoors. ElectricCar: Further derived class from Car with attributes like batteryCapacity.

Design a hierarchy of employee types within a company.

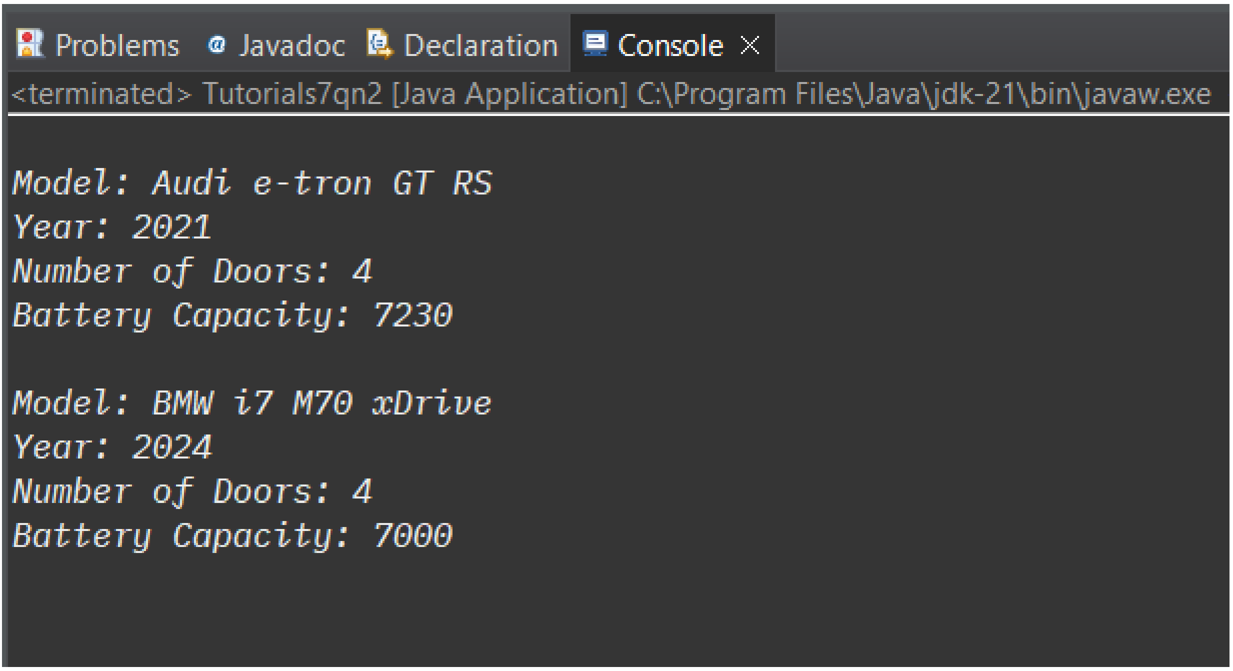


```java
/*
TUTORIALS 7
MODULE - 2
INHERITANCE IN JAVA

2. Write a Java program to implement Multilevel Inheritance
Implement a class hierarchy for different types of vehicles.
Classes:
Vehicle: Base class with attributes like model and year.
Car: Derived class from Vehicle with additional attributes like numberOfDoors.
ElectricCar: Further derived class from Car with attributes like batteryCapacity.
 Design a hierarchy of employee types within a company.

Karthik Krishnan
S3 CSE B
Roll: 45
*/
package Tutorials7;

class Vehicle {
    String Model;
    int Year;

    void vehicle() {
        System.out.println("\nModel: " + Model);
        System.out.println("Year: " + Year);
    }
}

class Car extends Vehicle {
    int No_of_Doors;

    void doors() {
        System.out.println("Number of Doors: " + No_of_Doors);
    }
}

class ElectricCar extends Car {
    int Battery_Capacity;

    ElectricCar(String Model, int Year, int No_of_Doors, int Battery_Capacity) {
        this.Model = Model;
        this.Year = Year;
        this.No_of_Doors = No_of_Doors;
        this.Battery_Capacity = Battery_Capacity;
    }

    void battery() {
        System.out.print("Battery Capacity: " + Battery_Capacity + "\n");
    }
}
```



```java

public class Tutorials7qn2{
    public static void main(String[] args) {
        ElectricCar c1 = new ElectricCar("Audi e-tron GT RS", 2021, 4, 7230);
        ElectricCar c2 = new ElectricCar("BMW i7 M70 xDrive", 2024, 4, 7000);
        c1.vehicle();
        c1.doors();
        c1.battery();

        c2.vehicle();
        c2.doors();
        c2.battery();
    }
}

```

OUTPUT:



<terminated> Tutorials7qn2 [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe

```

Model: Audi e-tron GT RS
Year: 2021
Number of Doors: 4
Battery Capacity: 7230

Model: BMW i7 M70 xDrive
Year: 2024
Number of Doors: 4
Battery Capacity: 7000
```

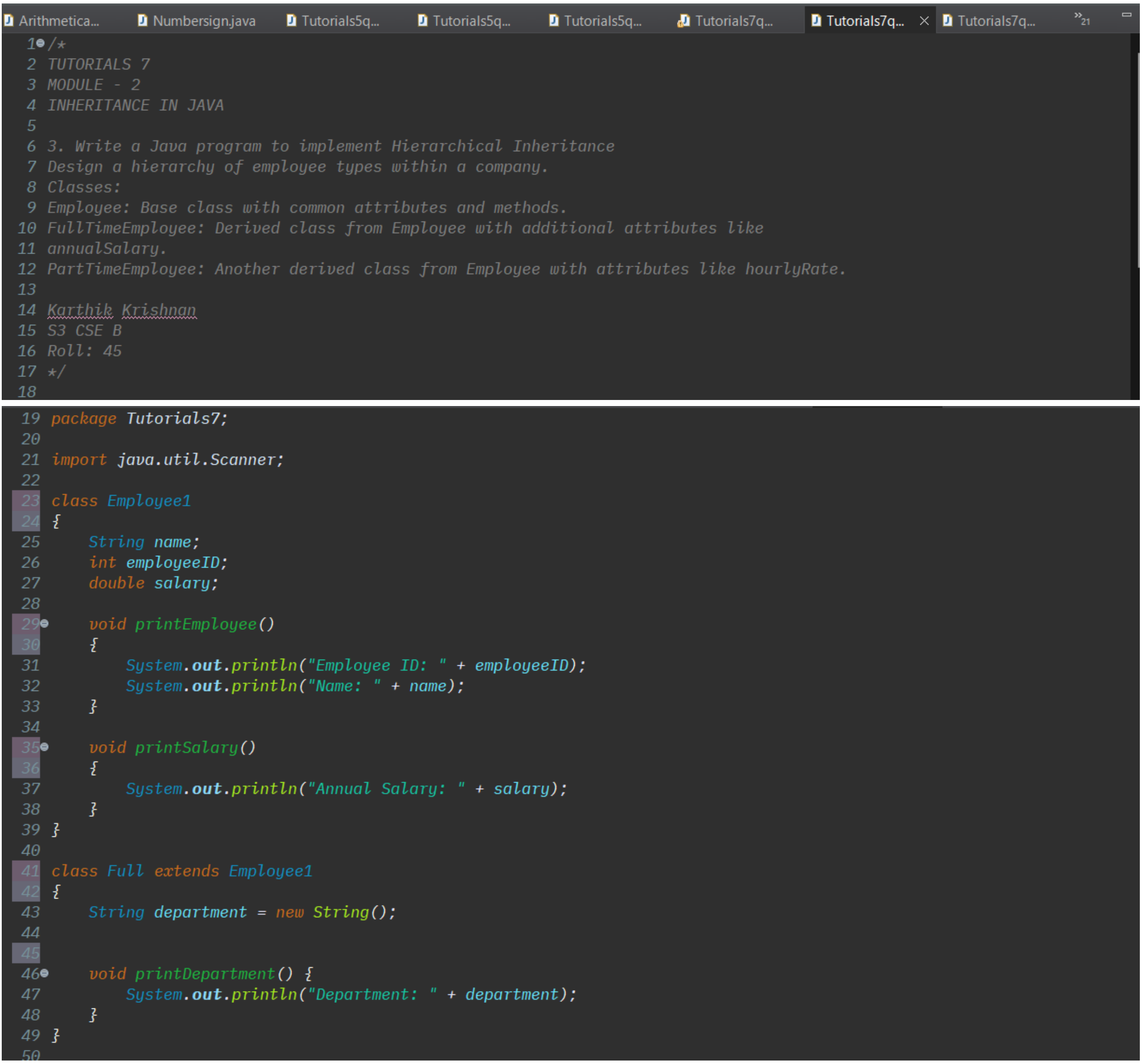
Qn 3) Write a Java program to implement Hierarchical Inheritance Design a hierarchy of employee types within a company.

Classes:

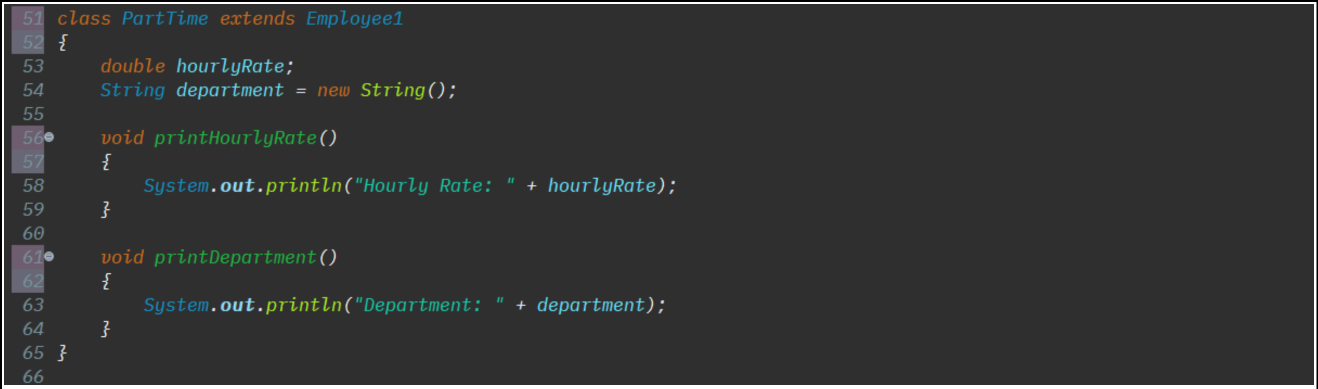
Employee: Base class with common attributes and methods.

FullTimeEmployee: Derived class from Employee with additional attributes like annualSalary.

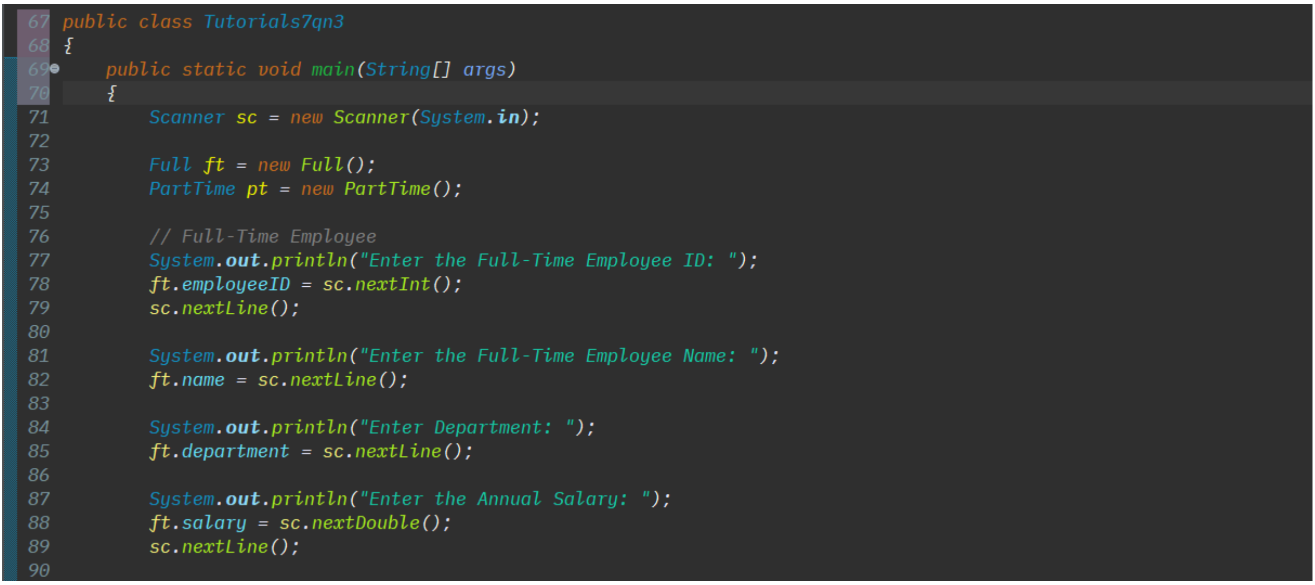
PartTimeEmployee: Another derived class from Employee with attributes like hourlyRate.



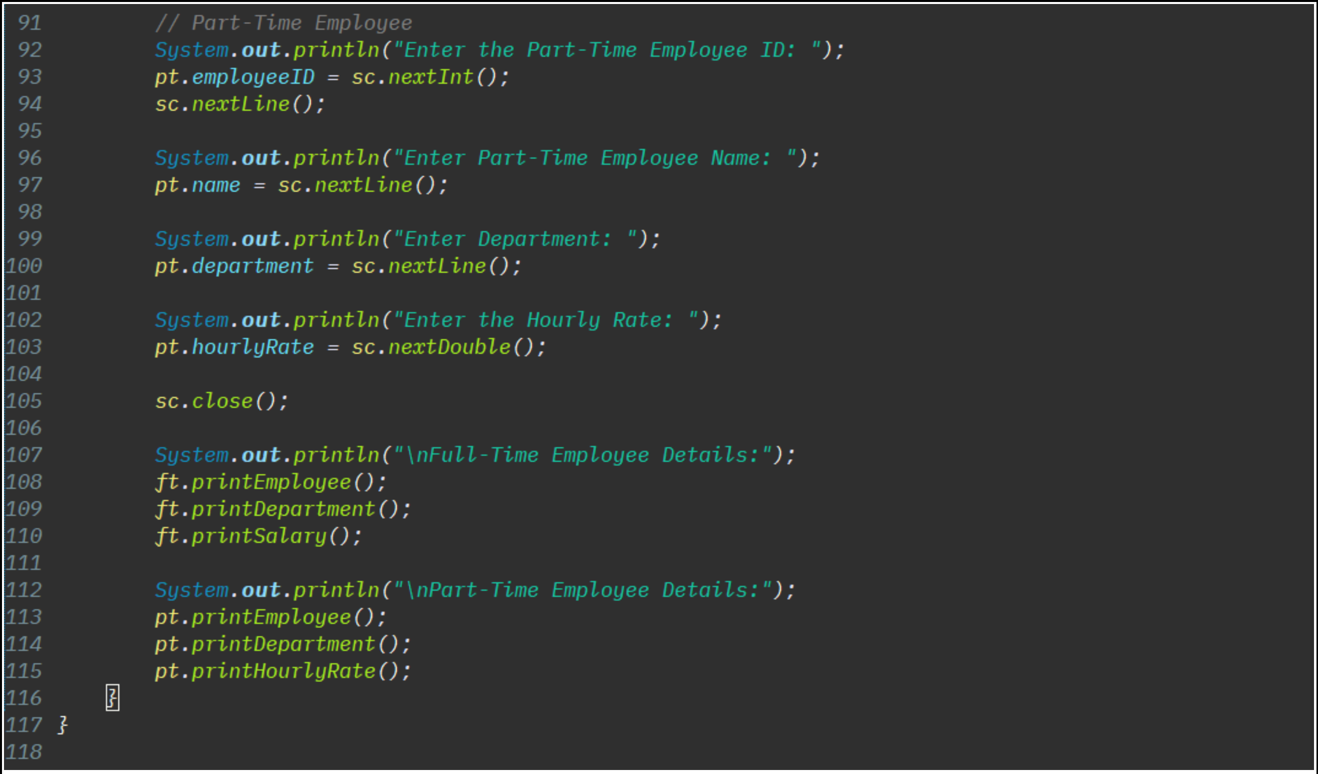
```java
/*
TUTORIALS 7
MODULE - 2
INHERITANCE IN JAVA

3. Write a Java program to implement Hierarchical Inheritance
Design a hierarchy of employee types within a company.
Classes:
Employee: Base class with common attributes and methods.
FullTimeEmployee: Derived class from Employee with additional attributes like
annualSalary.
PartTimeEmployee: Another derived class from Employee with attributes like hourlyRate.

Karthik Krishnan
S3 CSE B
Roll: 45
*/

package Tutorials7;

import java.util.Scanner;

class Employee1
{
    String name;
    int employeeID;
    double salary;

    void printEmployee()
    {
        System.out.println("Employee ID: " + employeeID);
        System.out.println("Name: " + name);
    }

    void printSalary()
    {
        System.out.println("Annual Salary: " + salary);
    }
}

class Full extends Employee1
{
    String department = new String();


    void printDepartment() {
        System.out.println("Department: " + department);
    }
}

```



```java
class PartTime extends Employee1
{
    double hourlyRate;
    String department = new String();

    void printHourlyRate()
    {
        System.out.println("Hourly Rate: " + hourlyRate);
    }

    void printDepartment()
    {
        System.out.println("Department: " + department);
    }
}

```

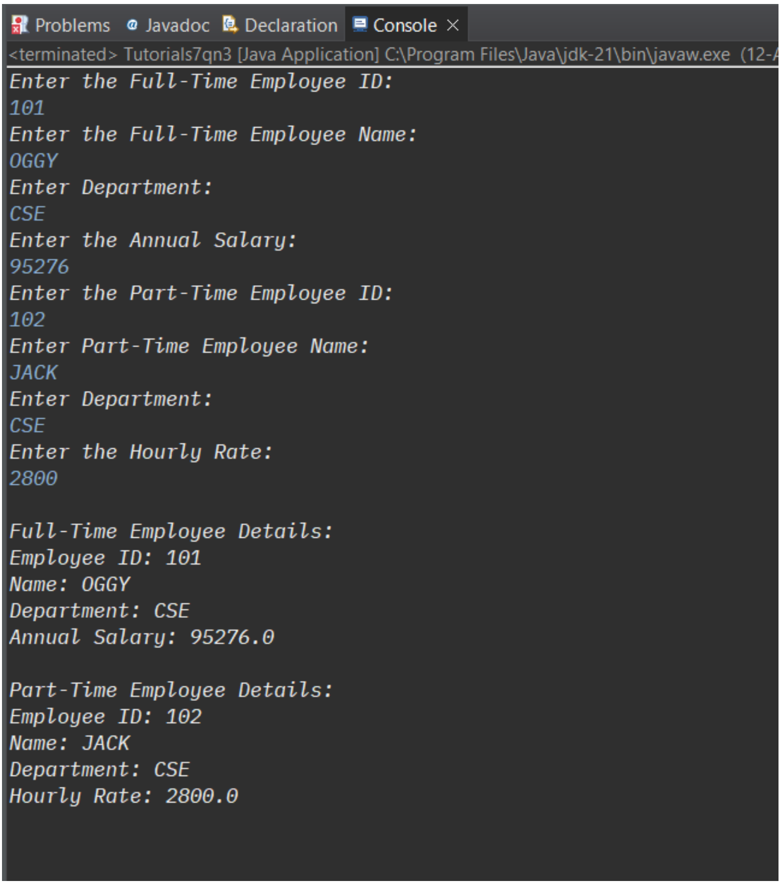


```java
public class Tutorials7qn3
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        Full ft = new Full();
        PartTime pt = new PartTime();

        // Full-Time Employee
        System.out.println("Enter the Full-Time Employee ID: ");
        ft.employeeID = sc.nextInt();
        sc.nextLine();

        System.out.println("Enter the Full-Time Employee Name: ");
        ft.name = sc.nextLine();

        System.out.println("Enter Department: ");
        ft.department = sc.nextLine();

        System.out.println("Enter the Annual Salary: ");
        ft.salary = sc.nextDouble();
        sc.nextLine();

```



```java
        // Part-Time Employee
        System.out.println("Enter the Part-Time Employee ID: ");
        pt.employeeID = sc.nextInt();
        sc.nextLine();

        System.out.println("Enter Part-Time Employee Name: ");
        pt.name = sc.nextLine();

        System.out.println("Enter Department: ");
        pt.department = sc.nextLine();

        System.out.println("Enter the Hourly Rate: ");
        pt.hourlyRate = sc.nextDouble();

        sc.close();

        System.out.println("\nFull-Time Employee Details:");
        ft.printEmployee();
        ft.printDepartment();
        ft.printSalary();

        System.out.println("\nPart-Time Employee Details:");
        pt.printEmployee();
        pt.printDepartment();
        pt.printHourlyRate();
    }
}
```

OUTPUT:



```
<terminated> Tutorials7qn3 [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (12-
Enter the Full-Time Employee ID: 
101
Enter the Full-Time Employee Name: 
OGGY
Enter Department: 
CSE
Enter the Annual Salary: 
95276
Enter the Part-Time Employee ID: 
102
Enter Part-Time Employee Name: 
JACK
Enter Department: 
CSE
Enter the Hourly Rate: 
2800

Full-Time Employee Details:
Employee ID: 101
Name: OGGY
Department: CSE
Annual Salary: 95276.0

Part-Time Employee Details:
Employee ID: 102
Name: JACK
Department: CSE
Hourly Rate: 2800.0
```