

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.

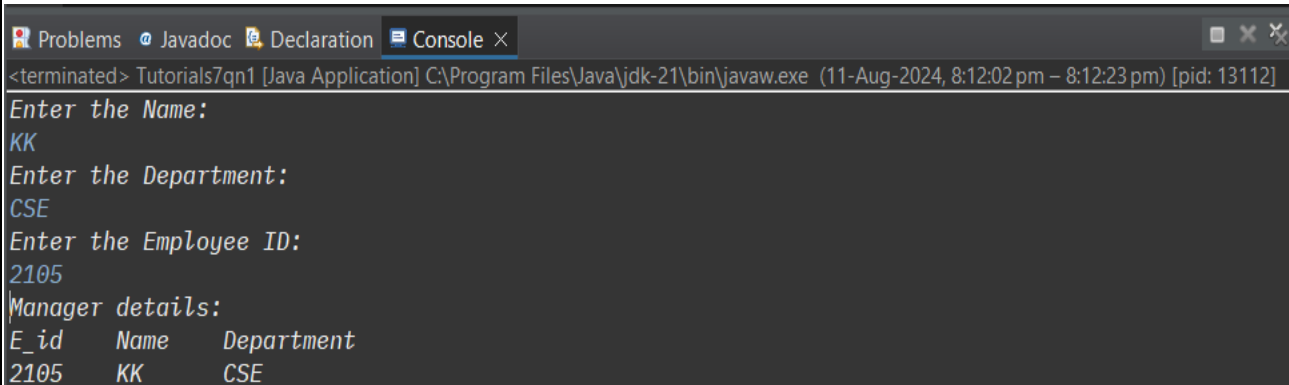
```
Arithmetica... Numbersign.java Tutorials5q... Tutorials6q... Tutorials6q... Tutorials7q... X Tutorials7q... Tutorials7q... »21
1  /*
2  TUTORIALS 7
3  MODULE - 2
4  INHERITANCE IN JAVA
5
6  1. Write a Java program to implement Single Inheritance
7  Create a simple application to manage employees and managers.
8  Classes:
9  Employee: Base class with attributes like name and employeeID.
10 Manager: Derived class that extends Employee, with additional attributes like
11 department and methods to manage employees.
12
13
14 Karthik Krishnan
15 S3 CSE B
16 Roll: 45
17 */
18
19 package Tutorials7;
20 import java.util.Scanner;
21
22 class Employee {
23     String name;
24     int E_id;
25 }
26
27 class Manager extends Employee {
28     String department;
29
30     void details() {
31         System.out.println(E_id + "\t" + name + "\t" + department);
32     }
33 }
```

```

34
35 public class Tutorial7qn1 {
36
37     public static void main(String[] args) {
38         Scanner sc = new Scanner(System.in);
39         Manager ma = new Manager();
40
41         System.out.println("Enter the Name: ");
42         ma.name = sc.nextLine();
43         System.out.println("Enter the Department: ");
44         ma.department = sc.nextLine();
45         System.out.println("Enter the Employee ID: ");
46         ma.E_id = sc.nextInt();
47
48         System.out.println("Manager details: ");
49         System.out.println("E_id\tName\tDepartment");
50         ma.details();
51     }
52 }

```

OUTPUT:



Problems Javadoc Declaration Console X

<terminated> Tutorial7qn1 [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.

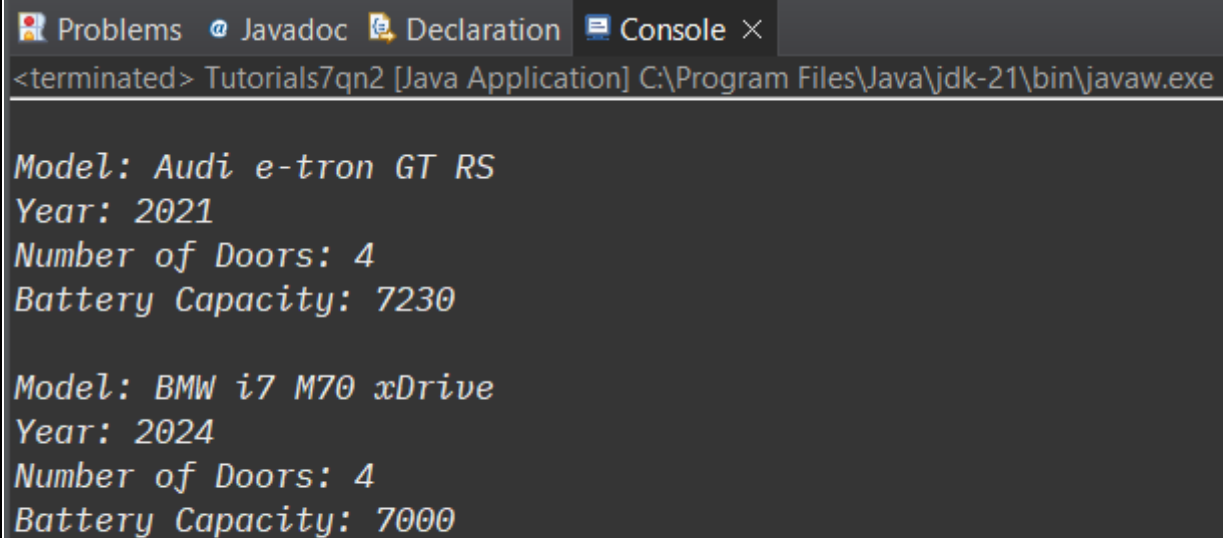
Car: Derived class from Vehicle with additional attributes like
numberOfDoors. ElectricCar: Further derived class from Car with
attributes like batteryCapacity.

Design a hierarchy of employee types within a company.

```
Arithmetica... Numbersign.java Tutorials5q... Tutorials5q... Tutorials5q... Tutorials5q... Tutorials7q... *Tutorials7q... x »21
1 /*
2 TUTORIALS 7
3 MODULE - 2
4 INHERITANCE IN JAVA
5
6 2. Write a Java program to implement Multilevel Inheritance
7 Implement a class hierarchy for different types of vehicles.
8 Classes:
9 Vehicle: Base class with attributes like model and year.
10 Car: Derived class from Vehicle with additional attributes like numberOfDoors.
11 ElectricCar: Further derived class from Car with attributes like batteryCapacity.
12 Design a hierarchy of employee types within a company.
13
14 Karthik Krishnan
15 S3 CSE B
16 Roll: 45
17 */
18
19 package Tutorials7;
20
21 class Vehicle {
22     String Model;
23     int Year;
24
25     void vehicle() {
26         System.out.println("\nModel: " + Model);
27         System.out.println("Year: " + Year);
28     }
29 }
30
31 class Car extends Vehicle {
32     int No_of_Doors;
33
34     void doors() {
35         System.out.println("Number of Doors: " + No_of_Doors);
36     }
37 }
38
39 class ElectricCar extends Car {
40     int Battery_Capacity;
41
42     ElectricCar(String Model, int Year, int No_of_Doors, int Battery_Capacity) {
43         this.Model = Model;
44         this.Year = Year;
45         this.No_of_Doors = No_of_Doors;
46         this.Battery_Capacity = Battery_Capacity;
47     }
48
49     void battery() {
50         System.out.print("Battery Capacity: " + Battery_Capacity + "\n");
51     }
52 }
```

```
53
54 public class Tutorials7qn2{
55     public static void main(String[] args) {
56         ElectricCar c1 = new ElectricCar("Audi e-tron GT RS", 2021, 4, 7230);
57         ElectricCar c2 = new ElectricCar("BMW i7 M70 xDrive", 2024, 4, 7000);
58         c1.vehicle();
59         c1.doors();
60         c1.battery();
61
62         c2.vehicle();
63         c2.doors();
64         c2.battery();
65     }
66 }
67
```

OUTPUT:



The screenshot shows an IDE console window with the following tabs: Problems, Javadoc, Declaration, and Console. The console title bar reads "<terminated> Tutorials7qn2 [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe". The output text is as follows:

```
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.

```
Arithmetica... Numbersign.java Tutorials5q... Tutorials5q... Tutorials5q... Tutorials7q... Tutorials7q... x Tutorials7q... »_21

1  /*
2  TUTORIALS 7
3  MODULE - 2
4  INHERITANCE IN JAVA
5
6  3. Write a Java program to implement Hierarchical Inheritance
7  Design a hierarchy of employee types within a company.
8  Classes:
9  Employee: Base class with common attributes and methods.
10 FullTimeEmployee: Derived class from Employee with additional attributes like
11 annualSalary.
12 PartTimeEmployee: Another derived class from Employee with attributes like hourlyRate.
13
14 Karthik Krishnan
15 S3 CSE B
16 Roll: 45
17 */
18

19 package Tutorials7;
20
21 import java.util.Scanner;
22
23 class Employee1
24 {
25     String name;
26     int employeeID;
27     double salary;
28
29     void printEmployee()
30     {
31         System.out.println("Employee ID: " + employeeID);
32         System.out.println("Name: " + name);
33     }
34
35     void printSalary()
36     {
37         System.out.println("Annual Salary: " + salary);
38     }
39 }
40
41 class Full extends Employee1
42 {
43     String department = new String();
44
45
46     void printDepartment() {
47         System.out.println("Department: " + department);
48     }
49 }
50
```

```
51 class PartTime extends Employee1
52 {
53     double hourlyRate;
54     String department = new String();
55
56     void printHourlyRate()
57     {
58         System.out.println("Hourly Rate: " + hourlyRate);
59     }
60
61     void printDepartment()
62     {
63         System.out.println("Department: " + department);
64     }
65 }
66
```

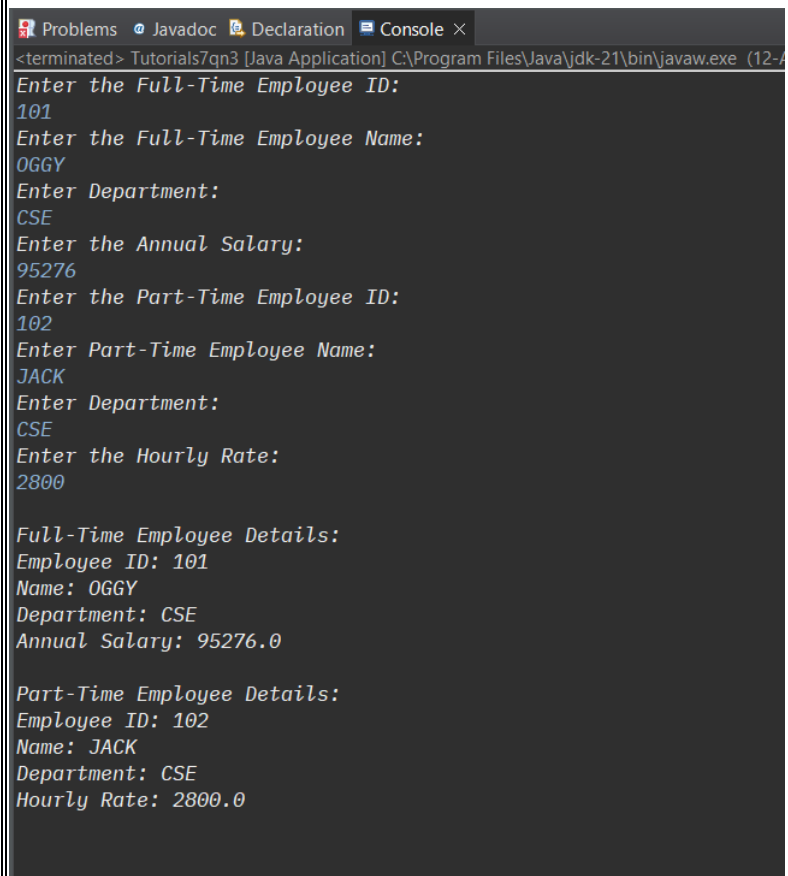
```
67 public class Tutorials7qn3
68 {
69     public static void main(String[] args)
70     {
71         Scanner sc = new Scanner(System.in);
72
73         Full ft = new Full();
74         PartTime pt = new PartTime();
75
76         // Full-Time Employee
77         System.out.println("Enter the Full-Time Employee ID: ");
78         ft.employeeID = sc.nextInt();
79         sc.nextLine();
80
81         System.out.println("Enter the Full-Time Employee Name: ");
82         ft.name = sc.nextLine();
83
84         System.out.println("Enter Department: ");
85         ft.department = sc.nextLine();
86
87         System.out.println("Enter the Annual Salary: ");
88         ft.salary = sc.nextDouble();
89         sc.nextLine();
90
```

```

91      // Part-Time Employee
92      System.out.println("Enter the Part-Time Employee ID: ");
93      pt.employeeID = sc.nextInt();
94      sc.nextLine();
95
96      System.out.println("Enter Part-Time Employee Name: ");
97      pt.name = sc.nextLine();
98
99      System.out.println("Enter Department: ");
100     pt.department = sc.nextLine();
101
102     System.out.println("Enter the Hourly Rate: ");
103     pt.hourlyRate = sc.nextDouble();
104
105     sc.close();
106
107     System.out.println("\nFull-Time Employee Details:");
108     ft.printEmployee();
109     ft.printDepartment();
110     ft.printSalary();
111
112     System.out.println("\nPart-Time Employee Details:");
113     pt.printEmployee();
114     pt.printDepartment();
115     pt.printHourlyRate();
116
117 }
118

```

OUTPUT:



The screenshot shows a Java IDE console window with the following output:

```

<terminated> Tutorials7qn3 [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (12-A
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

```