

**SAP Id: 500091584**  
**Roll No: R2142210822**  
**Name: Ujjwal Kumar Gupta**

**Course: CSE DevOps (Hons.)**  
**Program & Sem: B.tech & 3rd**  
**Session : 2021-25**

## **INDEX**

<b>S.No.</b>	<b>Experiment No.</b>	<b>Title</b>	<b>Date of Performance</b>	<b>Date of Submission</b>	<b>Remarks (By Faculty)</b>

Note: 1. Submit your original work otherwise you will lose marks.

2. Submit all your experiments into a single file

3. The submission must be properly compiled and in provided format only

## Experiment No. \_\_4\_\_

**Date of performance: 14/09/2022**

**Date of Submission: 14/09/2022**

**SAP Id: 500091584**

**Roll No.: R2142210822**

**Name of the Student: Ujjwal Kumar Gupta**

---

**1. Title:** Inheritance

**2. Objective:** Inheritance

**3. List of lab activities:** 1) Write a Java program to show that private member of a super class cannot be accessed from derived classes.

2) Write a program in Java to create a Player class. Inherit the classes Cricket \_Player, Football\_Player and Hockey\_ Player from Player class.

3) Write a class Worker and derive classes DailyWorker and SalariedWorker from it. Every worker has a name and a salary rate. Write method ComPay (int hours) to compute the week pay of every worker. A Daily Worker is paid on the basis of the number of days he/she works. The SalariedWorker gets paid the wage for 40 hours a week no matter what the actual hours are. Test this program to calculate the pay of workers. You are expected to use the concept of polymorphism to write this program.

4) Consider the trunk calls of a telephone exchange. A trunk call can be ordinary, urgent or lightning. The charges depend on the duration and the type of the call. Write a program using the concept of polymorphism in Java to calculate the charges.

5) Design a class employee of an organization. An employee has a name, empid, and salary. Write the default constructor, a constructor with parameters (name, empid, and salary) and methods to return name and salary. Also write a method increaseSalary that raises the employee's salary by a certain user specified percentage. Derive a subclass Manager from employee. Add an instance variable named department to the manager class. Supply a test program that uses these classes and methods.

#### 4. Algorithm/Flowchart and Code followed by Output screenshot (2samples for each program):

```
src > Experiment 4 > J derived.java > ...
1 class base{
2     private int a= 10;
3     private void sum()
4     {
5         int ans= a+a;
6         System.out.println("sum : "+ans);
7     }
8     void test()
9     {
10        System.out.println(a);
11    }
12 }
13
14 class child extends base{
15     int b=20;
16     void test();
17     void sum();
18 }
19
20 public class derived {
21
22     Run | Debug
23     public static void main(String[] args) {
24         child c1= new child();
25         c1.test();
26     }
27 }
```

```
}
derived.java:16: error: missing method body, or declare abstract
void test();
^
derived.java:17: error: missing method body, or declare abstract
void sum();
^
2 errors
PS C:\Users\Ujjwal\Documents\starting\src\Experiment 4> []
```

```
src > Experiment 4 > J sports.java > players > details()
1 import java.util.*;
2
3 class players {
4     String name; int age; float height;
5
6     void details()
7
8     {
9
10        Scanner sc = new Scanner(System.in);
11
12        System.out.println("name of the player :");
13        sc.nextLine();
14        System.out.println("age of the player :");
15        sc.nextInt();
16        System.out.println("height of the player :");
17        sc.nextFloat();
18        sc.close();
19    }
20 }
21
22 class cricket_player extends players {
23     float run_rate, strike_rate;
24
25     void cricketer_details(float run_rate, float Strike_rate) {
26         details();
27         System.out.println("run rate : " + run_rate);
28         System.out.println("strike rate : " + strike_rate);
29     }
30 }
```

```
src > Experiment 4 > J sports.java > players > details()
31 }
32
33 class football_player extends players {
34     void footballer_details(int goals, String position) {
35         details();
36         System.out.println("total number of goals : " + goals);
37         System.out.println("position : " + position);
38     }
39 }
40
41
42 class hockey_player extends players {
43
44     void hockey_details(int goals, String position) {
45         details();
46         System.out.println("total number of goals : " + goals);
47         System.out.println("position : " + position);
48     }
49 }
50
51 public class sports {
52     Run | Debug
53     public static void main(String[] args) {
54         cricket_player c1 = new cricket_player();
55         football_player f1 = new football_player();
56         hockey_player h1 = new hockey_player();
57
58         Scanner s1 = new Scanner(System.in);
59         System.out.println("enter 1 for cricket, 2 for football and 3 for hockey:");
```

```
Run Terminal Help sports.java - starting - Visual Studio Code
J derived.java 3 J sports.java X
src > Experiment 4 > J sports.java > ? players > details()
50
51 public class sports {
52     Run | Debug
53     public static void main(String[] args) {
54         cricket_player c1 = new cricket_player();
55         football_player f1 = new football_player();
56         hockey_player h1 = new hockey_player();
57
58         Scanner s1 = new Scanner(System.in);
59         System.out.println(x: "enter 1 for cricket, 2 for football and 3 for hockey:");
60
61         int a1 = s1.nextInt();
62         if (a1 == 1) {
63             c1.cricketer_details(run_rate: 20, Strike_rate: 40);
64         }
65         if (a1 == 2) {
66             f1.footballer_details(goals: 250, position: "striker");
67         }
68         if (a1 == 3)
69
70         {
71             h1.hockey_details(goals: 230, position: "striker");
72         }
73         s1.close();
74     }
75 }
76
77
```

```
PS C:\Users\Ujjwal\Documents\starting\src\Experiment 4> cd
> cd "C:\Users\Ujjwal\Documents\starting\src\Experiment 4"
enter 1 for cricket, 2 for football and 3 for hockey:
2
name of the player :
ujjwal
age of the player :
19
height of the player :
162
total number of goals : 250
position : striker
PS C:\Users\Ujjwal\Documents\starting\src\Experiment 4> |
```

```
Run Terminal Help wages.java - starting - Visual Studio Code
J wages.java 4 X
src > Experiment 4 > J wages.java > ? dailyworker > company()
1 import java.util.*;
2 class worker
3 {
4     String name;
5     float details()
6 {
7     Scanner s1= new Scanner(System.in);
8     System.out.println(x: "name of employee :");
9     String name= s1.nextLine();
10    System.out.println(x: "number of hours : ");
11
12    float h= s1.nextFloat();
13
14    return h;
15 }
16 }
17 class dailyworker extends worker{
18
19     // int rate=75;
20     void company()
21 {
22     float h= details();
23     Scanner s= new Scanner(System.in);
24     System.out.println(x: "enter the rate : ");
25     int rate;
26     rate= s.nextInt();
27
28
29     System.out.println("pay : "+ rate*h);
30 }
31 }
32 class salariedworker extends worker
```

```
J wages.java 4 X
src > Experiment 4 > J wages.java > ? dailyworker > company()
32 class salariedworker extends worker
33 {
34     // int rate=100;
35     // int hour=40;
36     void company()
37 {
38     float h = details();
39     Scanner s= new Scanner(System.in);
40
41     System.out.println(x: "enter the rate : ");
42     int rate;
43     rate= s.nextInt();
44     if(h<40)
45     {
46         System.out.println("pay : "+ rate);
47     }
48 }
49
50
51
52 public class wages {
53     Run | Debug
54     public static void main(String[] args) {
55         // worker w1= new worker();
56         // w1.details();
57         dailyworker d1= new dailyworker();
58         salariedworker s1= new salariedworker();
59         d1.company();
60         s1.company();
61     }
62 }
```

```
PS C:\Users\Ujjwal\Documents\starting\src\Experiment 4> cd "C:\Users\Ujjwal\Documents\starting\src\Experiment 4"
name of employee :
ujjwal
number of hours :
40
enter the rate :
2
pay : 92.0
name of employee :
u
number of hours :
20
enter the rate :
20
pay : 20
PS C:\Users\Ujjwal\Documents\starting\src\Experiment 4> |
```

```
trunkcall.java - starting - Visual Studio Code
J trunkcall.java X
src > Experiment 4 > J trunkcall.java > ...
1 class ordinary {
2     int mins = 15;
3     int rate = 2;
4
5     void pay() {
6         System.out.println("charges for ordinary call : " + mins * rate);
7     }
8 }
9
10
11 class urgent {
12     int mins = 15;
13     float rate = 5.5f;
14
15     void pay() {
16         System.out.println("charges for urgent call : " + mins * rate);
17     }
18 }
19
20
21
22
23 class lightning {
24     int mins = 15;
25     int rate = 10;
26
27     void pay() {
28         System.out.println("charges for lightning call : " + mins * rate);
29     }
30 }
31
32
33 public class trunkcall {
34     Run | Debug
35     public static void main(String[] args) {
36         ordinary o1 = new ordinary();
37         urgent u1 = new urgent();
38         lightning l1 = new lightning();
39         l1.pay();
40     }
41 }
42
43
44
```

```
trunkcall.java - starting - Visual Studio Code
J trunkcall.java X
src > Experiment 4 > J trunkcall.java > ...
30 }
31 }
32 }
33 public class trunkcall {
34     Run | Debug
35     public static void main(String[] args) {
36         ordinary o1 = new ordinary();
37         o1.pay();
38         urgent u1 = new urgent();
39         u1.pay();
40         lightning l1 = new lightning();
41         l1.pay();
42     }
43 }
44
```

```
OUTPUT TERMINAL JUPYTER DEBUG CONSOLE
PS C:\Users\Ujjwal\Documents\starting\src\Experiment 4> cd "C:\Users\Ujjwal\Documents\starting\src\Experiment 4"
all }
charges for ordinary call : 30
charges for urgent call : 82.5
charges for lightning call : 150
PS C:\Users\Ujjwal\Documents\starting\src\Experiment 4>
```

```
employeesalary.java - starting - Visual Studio Code
J trunkcall.java J employeesalary.java X
src > Experiment 4 > J employeesalary.java > employee > employee(int, String, float)
1 class employee
2 {
3     String name;
4     int empid;
5     float salary;
6     employee() //default constructor
7     {
8     }
9
10    employee(int id, String n, float sal) //parameterized constructor
11    {
12        empid=id;
13        name=n;
14        salary=sal;
15    }
16
17    void details()
18    {
19        System.out.println("employee id : "+empid);
20        System.out.println("employee name : "+name);
21        System.out.println("salary of the employee : "+ salary);
22    }
23
24    void increasesalary()
25    {
26        float incsalary= salary+ (salary *20/100);
27        System.out.println("increase salary is : "+incsalary);
28    }
29
30    class manager extends employee
31    {
32        public String department= "d";
33    }
34
35    public class employeesalary
36    {
37
38
39
40
41
42
43
44
45
46
47
48
49
```

```
employeesalary.java - starting - Visual Studio Code
J trunkcall.java J employeesalary.java X
src > Experiment 4 > J employeesalary.java > employee > employee(int, String, float)
28 }
29 class manager extends employee
30 {
31
32     public String department= "d";
33 }
34
35 public class employeesalary
36 {
37     Run | Debug
38     public static void main(String[] args) {
39
40         employee e1= new employee(id: 24, n: "ujjwal",sal: 2500);
41         e1.details();
42         e1.increasesalary();
43     }
44 }
45
46
47
48
49
```

```
OUTPUT TERMINAL JUPYTER DEBUG CONSOLE
PS C:\Users\Ujjwal\Documents\starting\src\Experiment 4> cd "C:\Users\Ujjwal\Documents\starting\src\Experiment 4"
employeesalary }
employee id : 24
employee name : ujjwal
salary of the employee :2500.0
increase salary is :3000.0
PS C:\Users\Ujjwal\Documents\starting\src\Experiment 4>
```

## 5. Brief notes about all the concepts related to the lab experiment

Inheritance represents the **IS-A relationship** which is also known as a *parent-child* relationship.

```
class Subclass-name extends Superclass-name
{
    //methods and fields
}
```