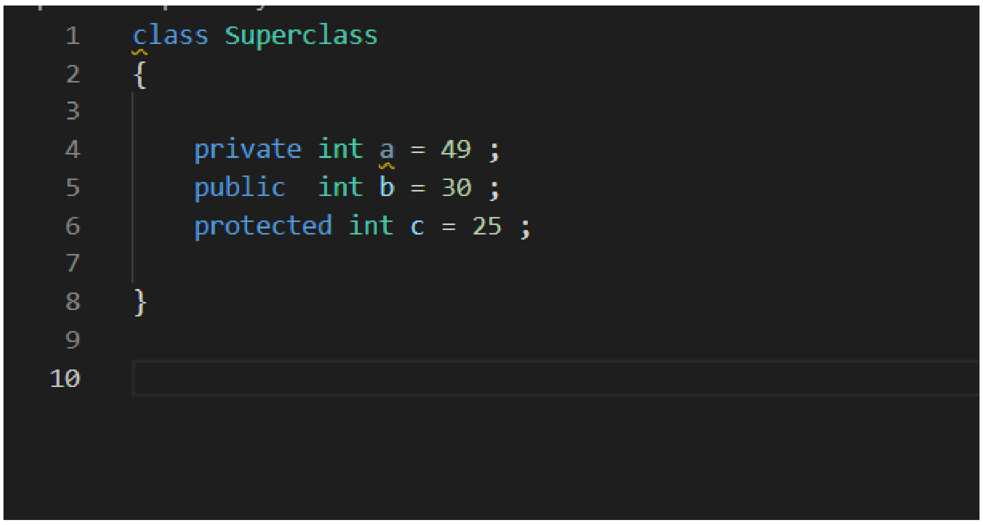
Lab Activity 4

Name – Sarvagya Gupta
Sap ID – 500083195
Roll No.- R2142202047

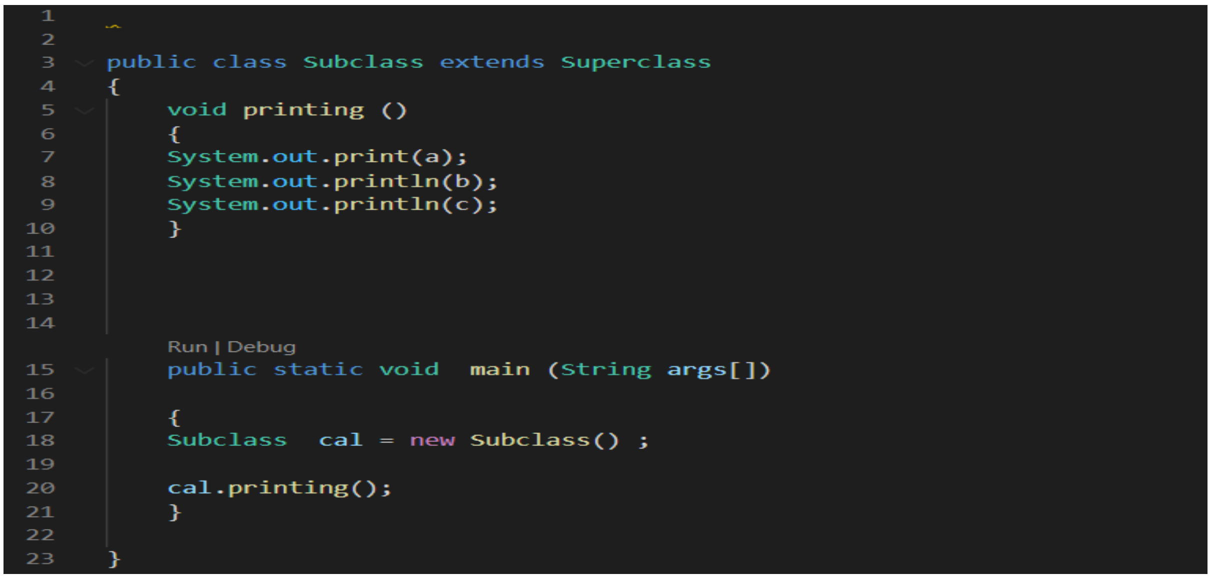
Title: Inheritance

1. Write a Java program to show that private members of a superclass cannot be accessed from derived classes.

Code

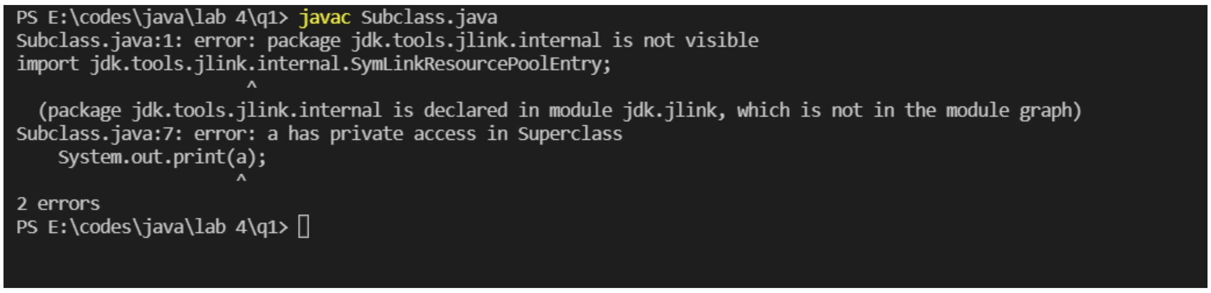


```java
class Superclass
{

    private int a = 49 ;
    public  int b = 30 ;
    protected int c = 25 ;

}

```



```java


public class Subclass extends Superclass
{
    void printing ()
    {
    System.out.print(a);
    System.out.println(b);
    System.out.println(c);
    }




    Run | Debug
    public static void  main (String args[])

    {
    Subclass  cal = new Subclass() ;

    cal.printing();
    }

}
```

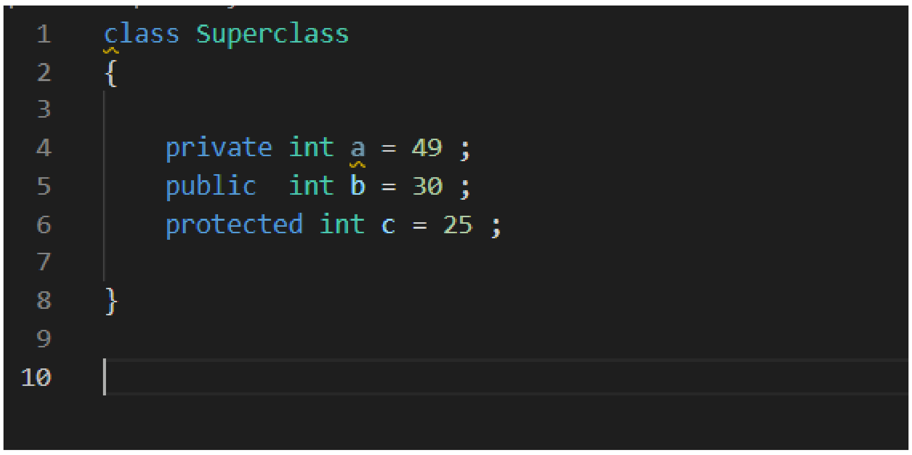
Output



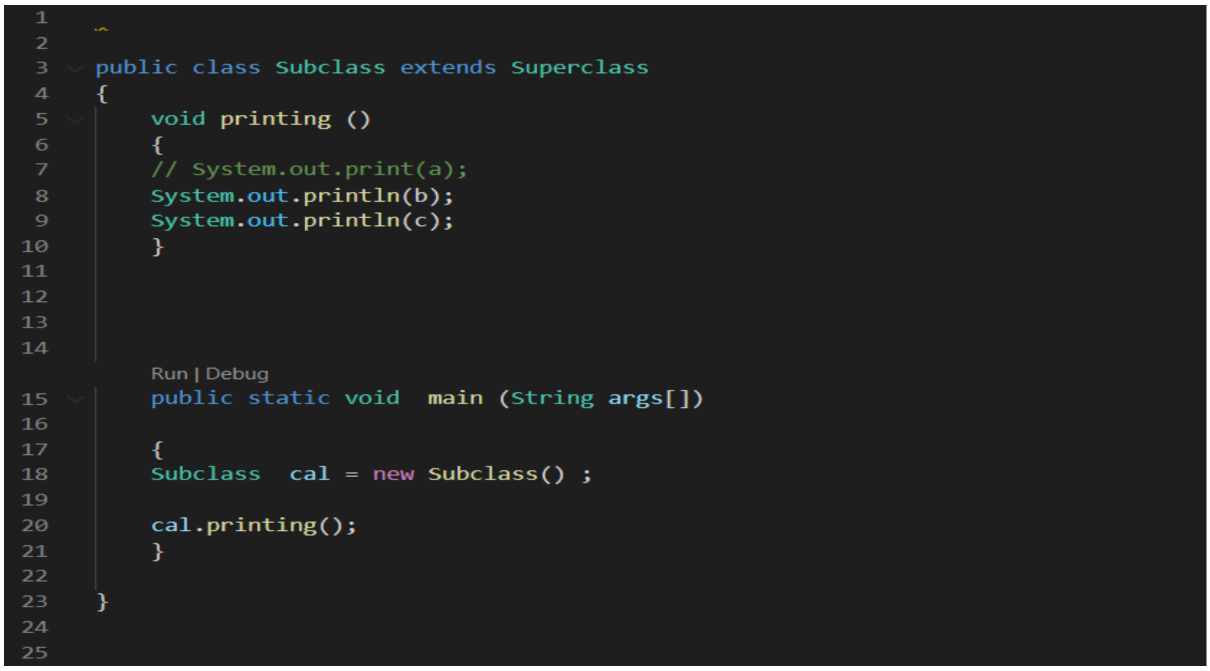
```
PS E:\codes\java\lab 4\q1> javac Subclass.java
Subclass.java:1: error: package jdk.tools.jlink.internal is not visible
import jdk.tools.jlink.internal.SymLinkResourcePoolEntry;
                      ^
  (package jdk.tools.jlink.internal is declared in module jdk.jlink, which is not in the module graph)
Subclass.java:7: error: a has private access in Superclass
    System.out.print(a);
                     ^
2 errors
PS E:\codes\java\lab 4\q1> 
```

If we remove the comment on the print statements of the private variables.

Code

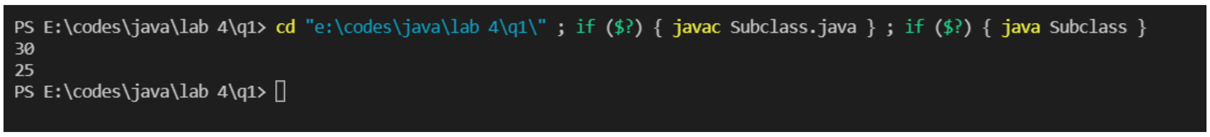


```
class Superclass
{

    private int a = 49 ;
    public  int b = 30 ;
    protected int c = 25 ;

}

```



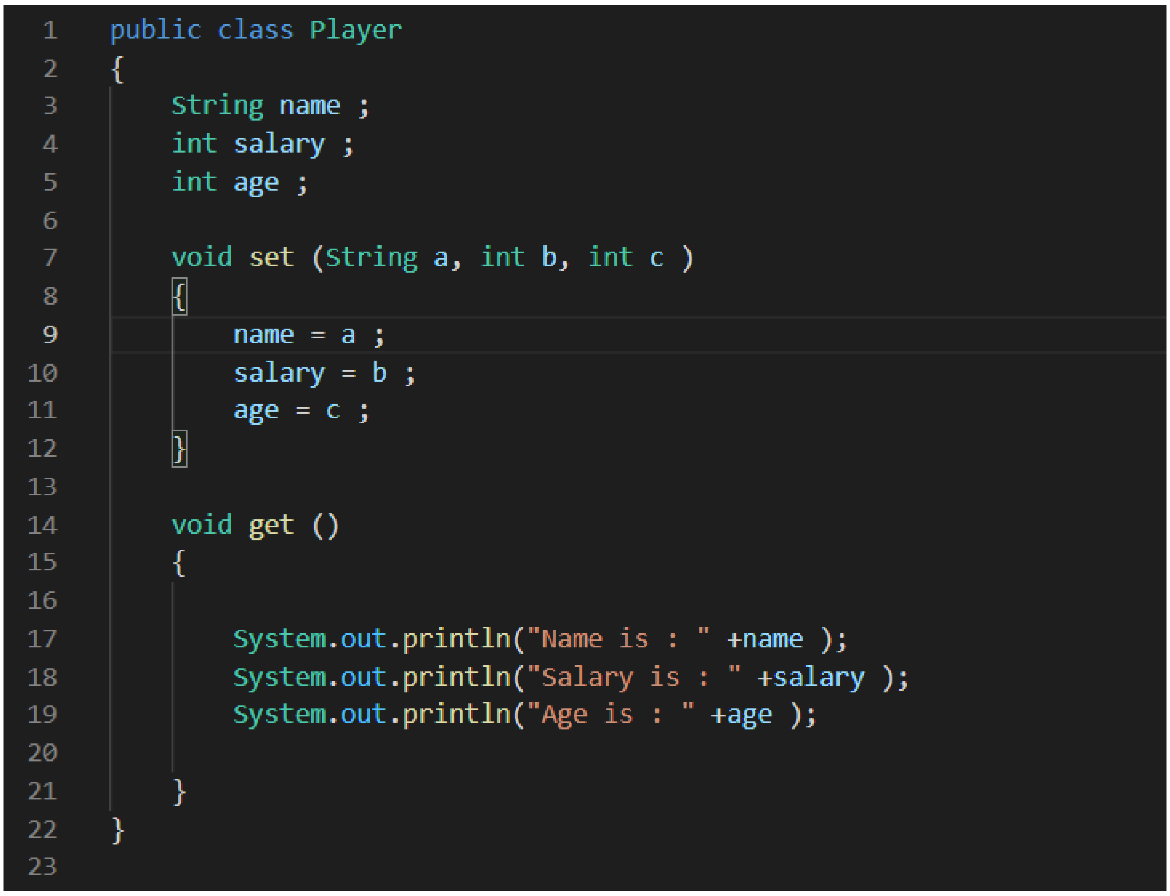
```


public class Subclass extends Superclass
{
    void printing ()
    {
    // System.out.print(a);
    System.out.println(b);
    System.out.println(c);
    }




    Run | Debug
    public static void  main (String args[])

    {
    Subclass  cal = new Subclass() ;

    cal.printing();
    }

}
```

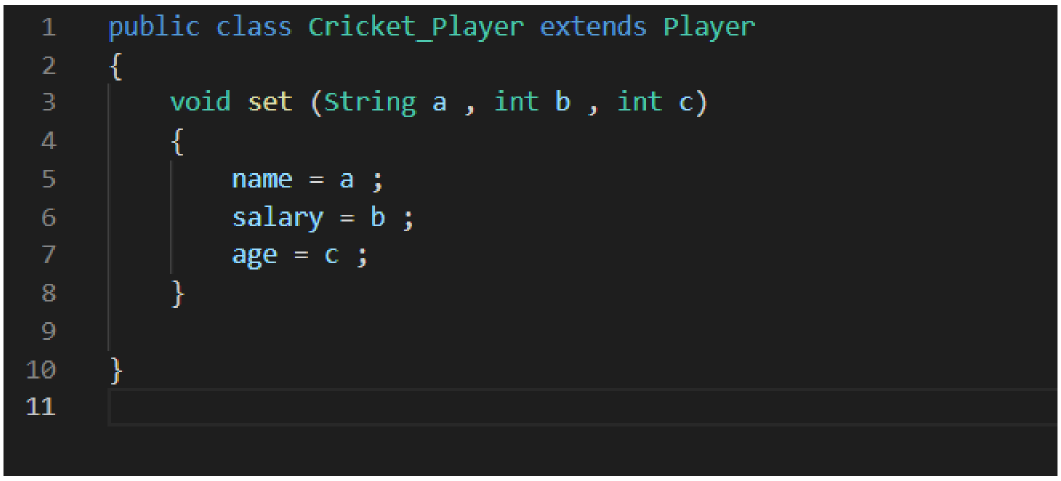
Output


```
PS E:\codes\java\lab 4\q1> cd "e:\codes\java\lab 4\q1\" ; if ($?) { javac Subclass.java } ; if ($?) { java Subclass }
30
25
PS E:\codes\java\lab 4\q1> 
```

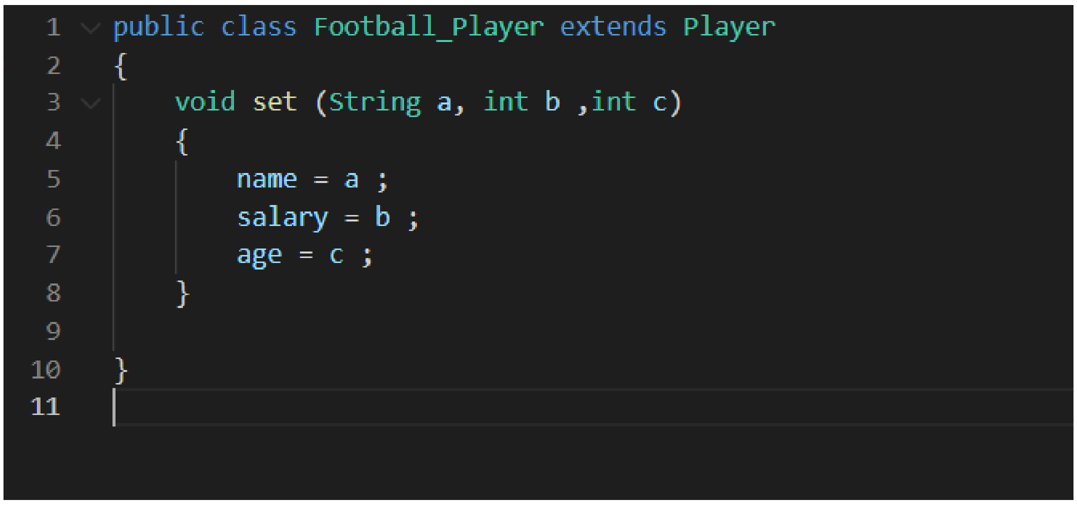
2. Write a program in Java to create a Player class. Inherit the classes Cricket _Player, Football _Player, and Hockey_ Player from Player class.

Code

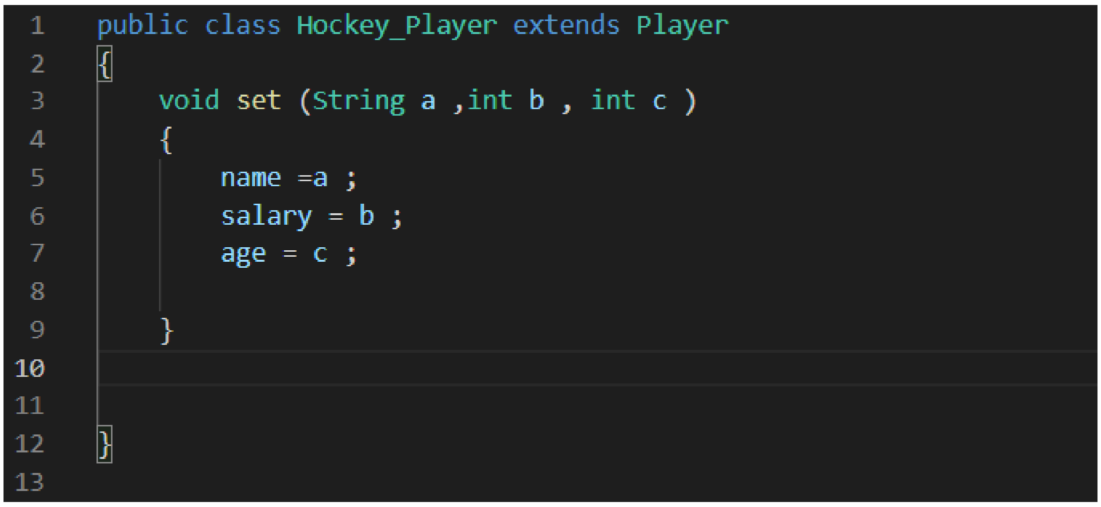

```java
public class Player
{
    String name ;
    int salary ;
    int age ;

    void set (String a, int b, int c )
    {
        name = a ;
        salary = b ;
        age = c ;
    }

    void get ()
    {

        System.out.println("Name is : " +name );
        System.out.println("Salary is : " +salary );
        System.out.println("Age is : " +age );

    }
}

```



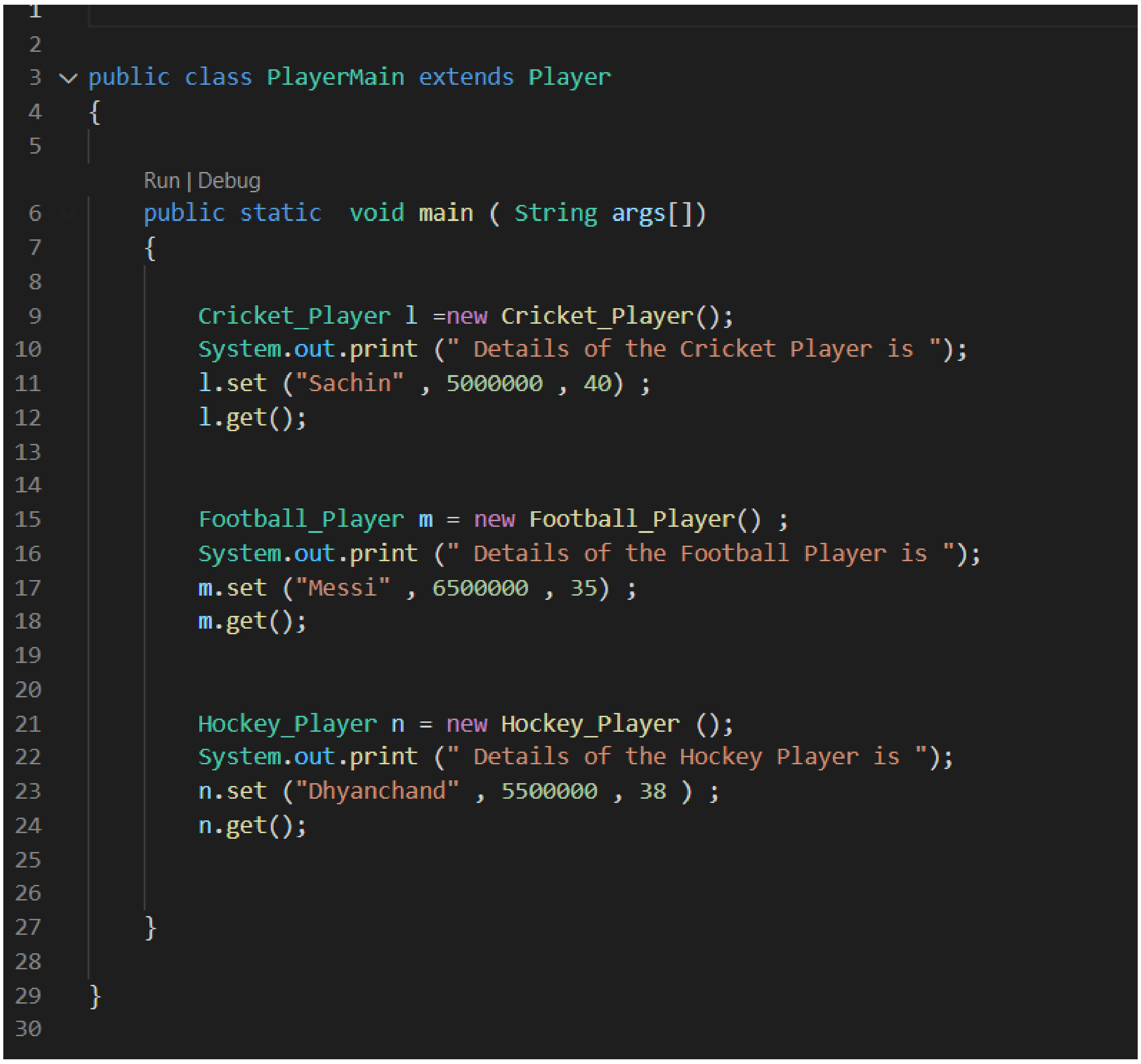
```java
public class Cricket_Player extends Player
{
    void set (String a , int b , int c)
    {
        name = a ;
        salary = b ;
        age = c ;
    }

}

```



```java
public class Football_Player extends Player
{
    void set (String a, int b ,int c)
    {
        name = a ;
        salary = b ;
        age = c ;
    }

}

```

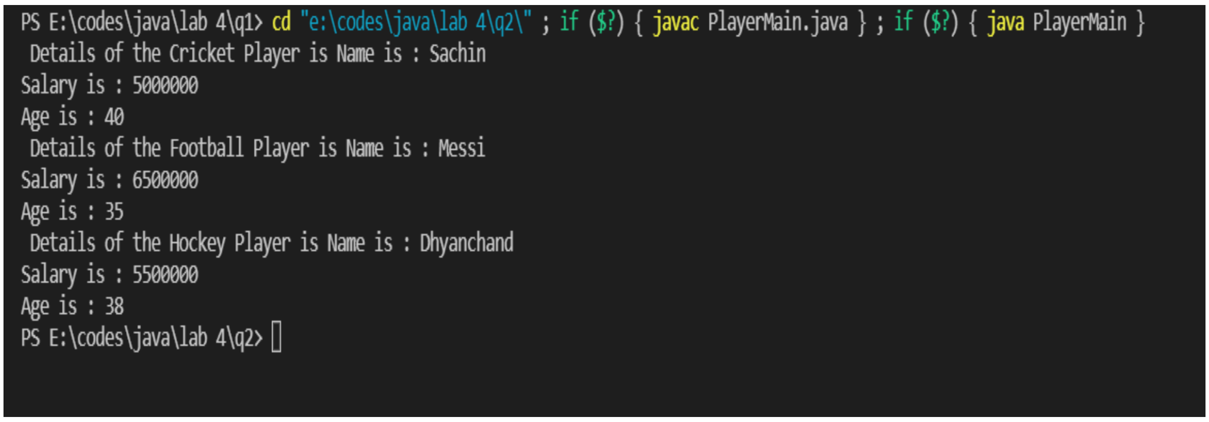


```java
public class Hockey_Player extends Player
{
    void set (String a ,int b , int c )
    {
        name =a ;
        salary = b ;
        age = c ;

    }


}

```



```java


public class PlayerMain extends Player
{

    Run | Debug
    public static  void main ( String args[])
    {

        Cricket_Player l =new Cricket_Player();
        System.out.print (" Details of the Cricket Player is ");
        l.set ("Sachin" , 5000000 , 40) ;
        l.get();


        Football_Player m = new Football_Player() ;
        System.out.print (" Details of the Football Player is ");
        m.set ("Messi" , 6500000 , 35) ;
        m.get();


        Hockey_Player n = new Hockey_Player ();
        System.out.print (" Details of the Hockey Player is ");
        n.set ("Dhyanchand" , 5500000 , 38 ) ;
        n.get();


    }

}

```

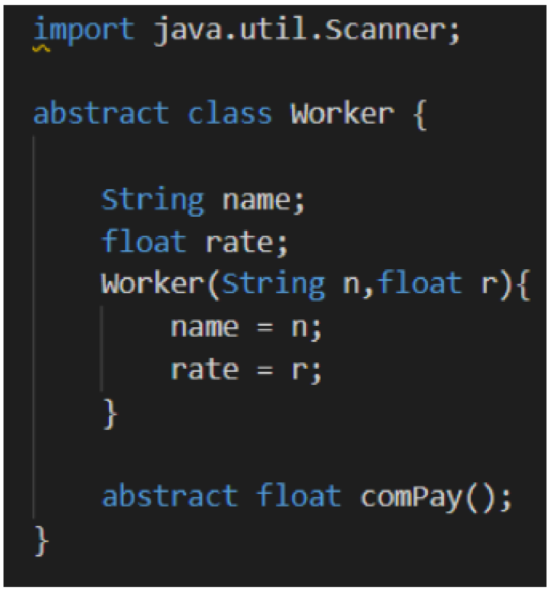
Output



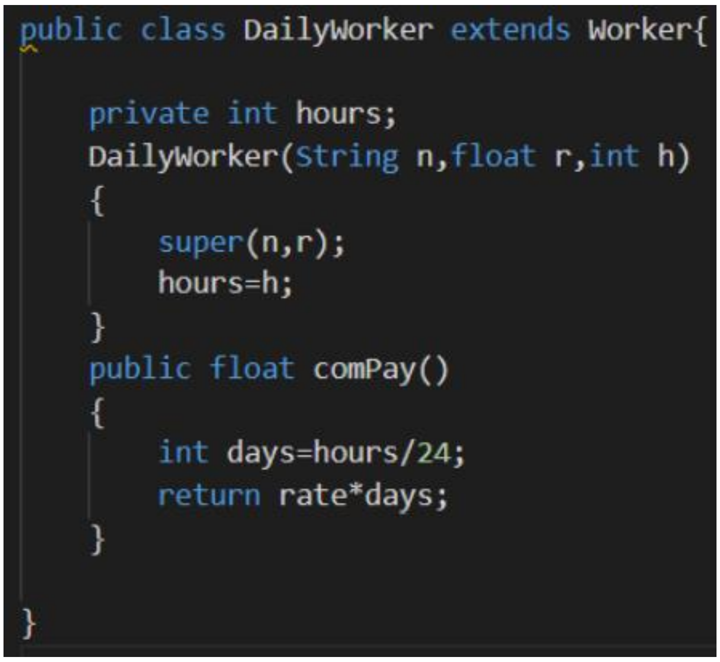
```
PS E:\codes\java\lab 4\q1> cd "e:\codes\java\lab 4\q2\" ; if ($?) { javac PlayerMain.java } ; if ($?) { java PlayerMain }
 Details of the Cricket Player is Name is : Sachin
Salary is : 5000000
Age is : 40
 Details of the Football Player is Name is : Messi
Salary is : 6500000
Age is : 35
 Details of the Hockey Player is Name is : Dhyanchand
Salary is : 5500000
Age is : 38
PS E:\codes\java\lab 4\q2> 
```

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 weekly pay of every worker. A Daily Worker is paid on the basis of the number of days he/she works. The Salaried Worker 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.

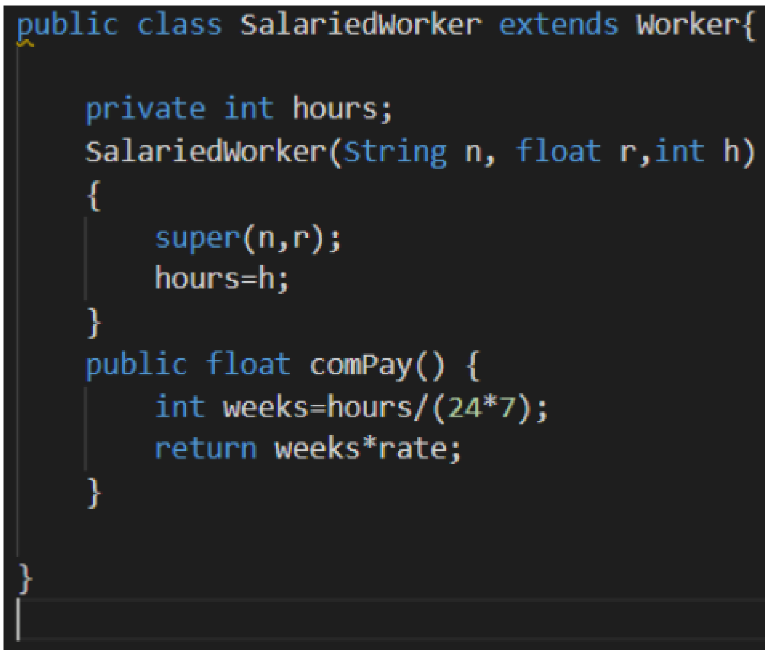
Code



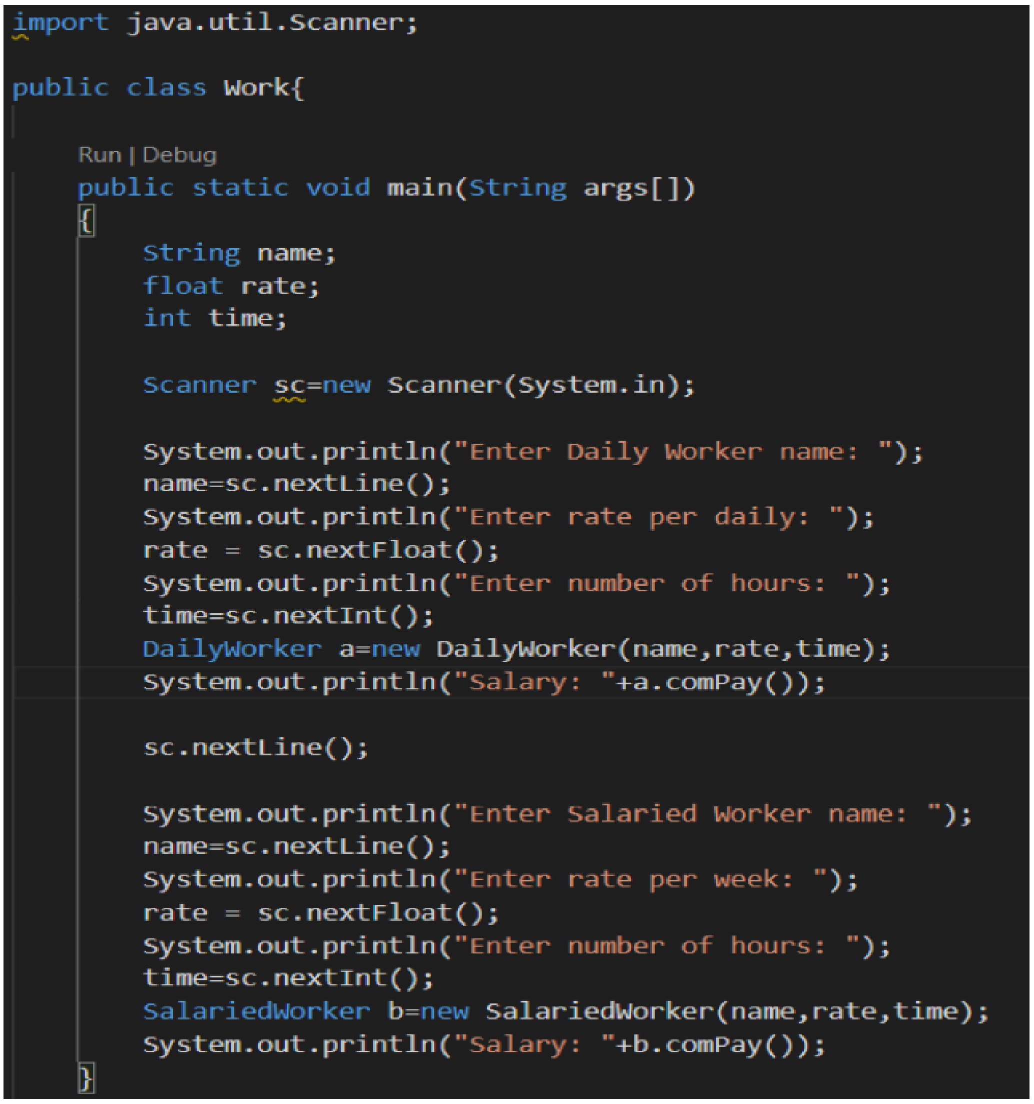
```java
import java.util.Scanner;

abstract class Worker {

    String name;
    float rate;
    Worker(String n,float r){
        name = n;
        rate = r;
    }

    abstract float comPay();
}
```



```java
public class DailyWorker extends Worker{

    private int hours;
    DailyWorker(String n,float r,int h)
    {
        super(n,r);
        hours=h;
    }
    public float comPay()
    {
        int days=hours/24;
        return rate*days;
    }

}
```

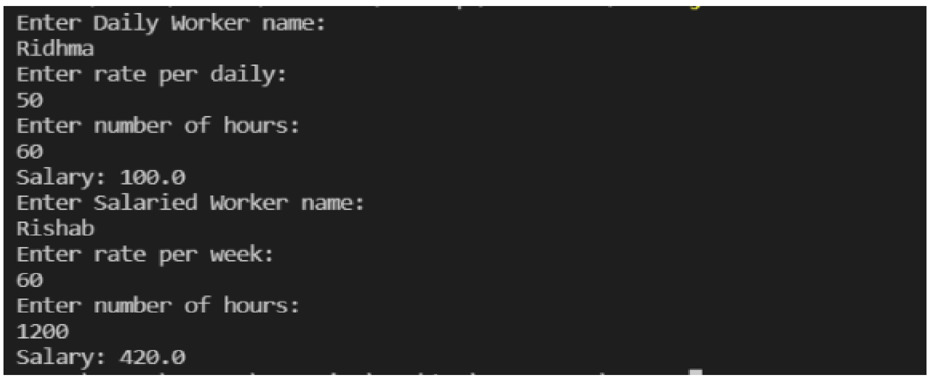


```java
public class SalariedWorker extends Worker{

    private int hours;
    SalariedWorker(String n, float r,int h)
    {
        super(n,r);
        hours=h;
    }
    public float comPay() {
        int weeks=hours/(24*7);
        return weeks*rate;
    }

}
```



```java
import java.util.Scanner;

public class Work{

    Run | Debug
    public static void main(String args[])
    {
        String name;
        float rate;
        int time;

        Scanner sc=new Scanner(System.in);

        System.out.println("Enter Daily Worker name: ");
        name=sc.nextLine();
        System.out.println("Enter rate per daily: ");
        rate = sc.nextFloat();
        System.out.println("Enter number of hours: ");
        time=sc.nextInt();
        DailyWorker a=new DailyWorker(name,rate,time);
        System.out.println("Salary: "+a.comPay());

        sc.nextLine();

        System.out.println("Enter Salaried Worker name: ");
        name=sc.nextLine();
        System.out.println("Enter rate per week: ");
        rate = sc.nextFloat();
        System.out.println("Enter number of hours: ");
        time=sc.nextInt();
        SalariedWorker b=new SalariedWorker(name,rate,time);
        System.out.println("Salary: "+b.comPay());
    }
```

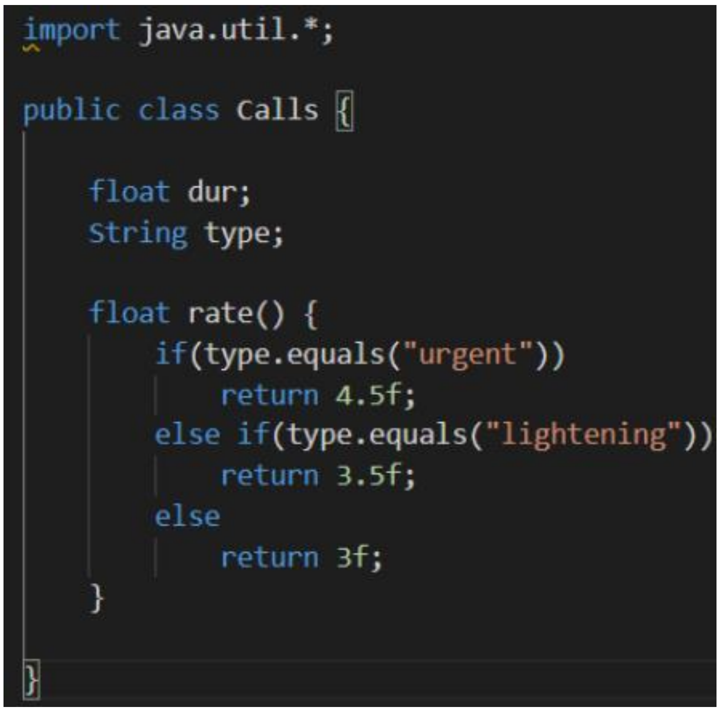
Output



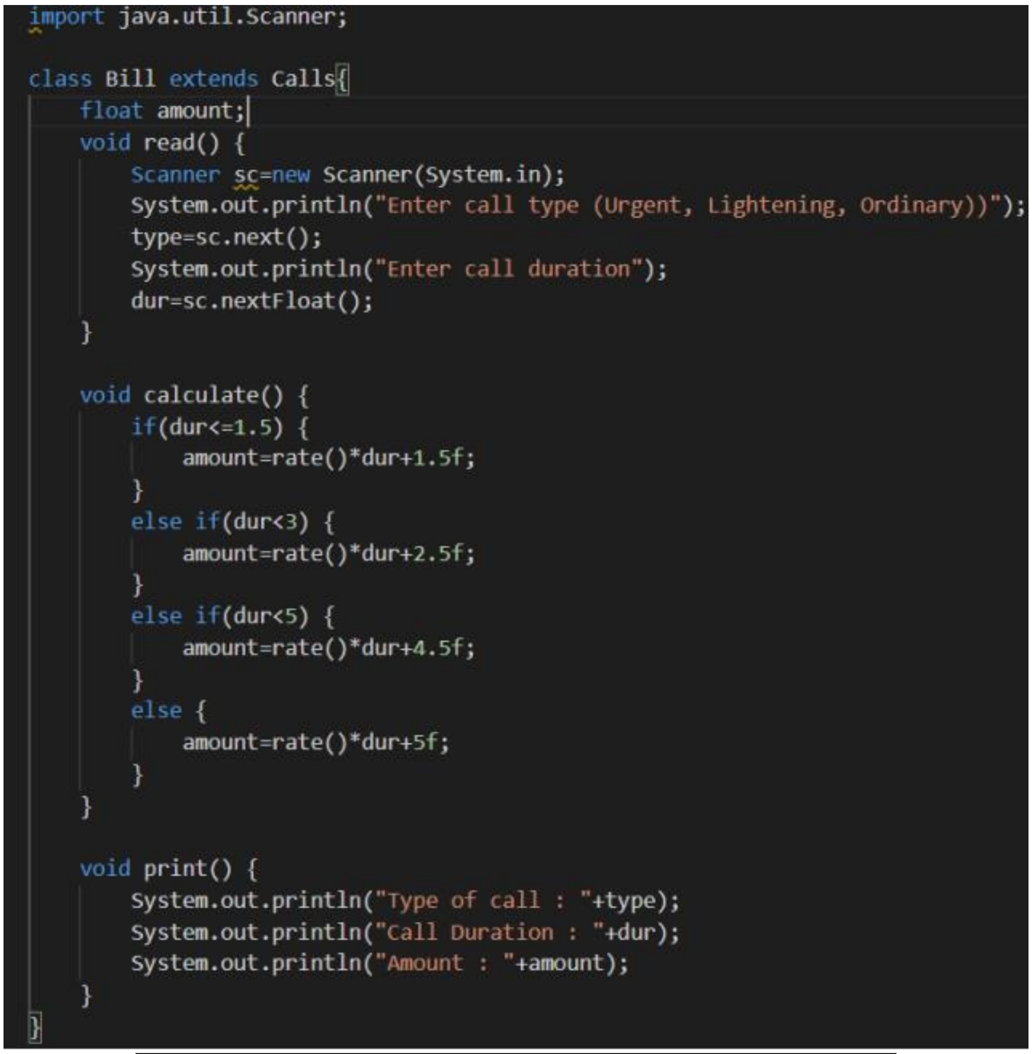
```
Enter Daily Worker name:
Ridhma
Enter rate per daily:
50
Enter number of hours:
60
Salary: 100.0
Enter Salaried Worker name:
Rishab
Enter rate per week:
60
Enter number of hours:
1200
Salary: 420.0
```

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.

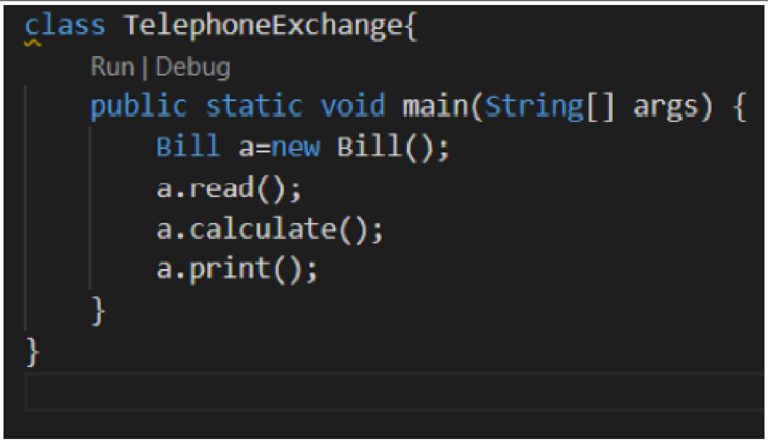
Code



```java
import java.util.*;

public class Calls {

    float dur;
    String type;

    float rate() {
        if(type.equals("urgent"))
            return 4.5f;
        else if(type.equals("lightening"))
            return 3.5f;
        else
            return 3f;
    }

}
```

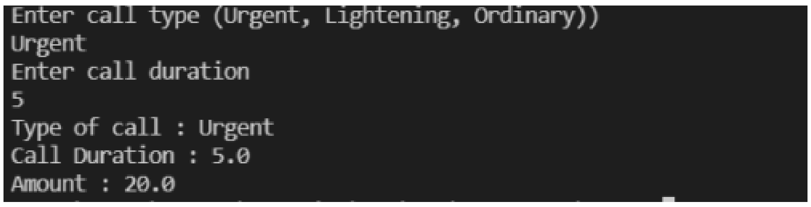


```java
import java.util.Scanner;

class Bill extends Calls{
    float amount;
    void read() {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter call type (Urgent, Lightening, Ordinary))");
        type=sc.next();
        System.out.println("Enter call duration");
        dur=sc.nextFloat();
    }

    void calculate() {
        if(dur<=1.5) {
            amount=rate()*dur+1.5f;
        }
        else if(dur<3) {
            amount=rate()*dur+2.5f;
        }
        else if(dur<5) {
            amount=rate()*dur+4.5f;
        }
        else {
            amount=rate()*dur+5f;
        }
    }

    void print() {
        System.out.println("Type of call : "+type);
        System.out.println("Call Duration : "+dur);
        System.out.println("Amount : "+amount);
    }
}
```



```java
class TelephoneExchange{
    Run | Debug
    public static void main(String[] args) {
        Bill a=new Bill();
        a.read();
        a.calculate();
        a.print();
    }
}
```

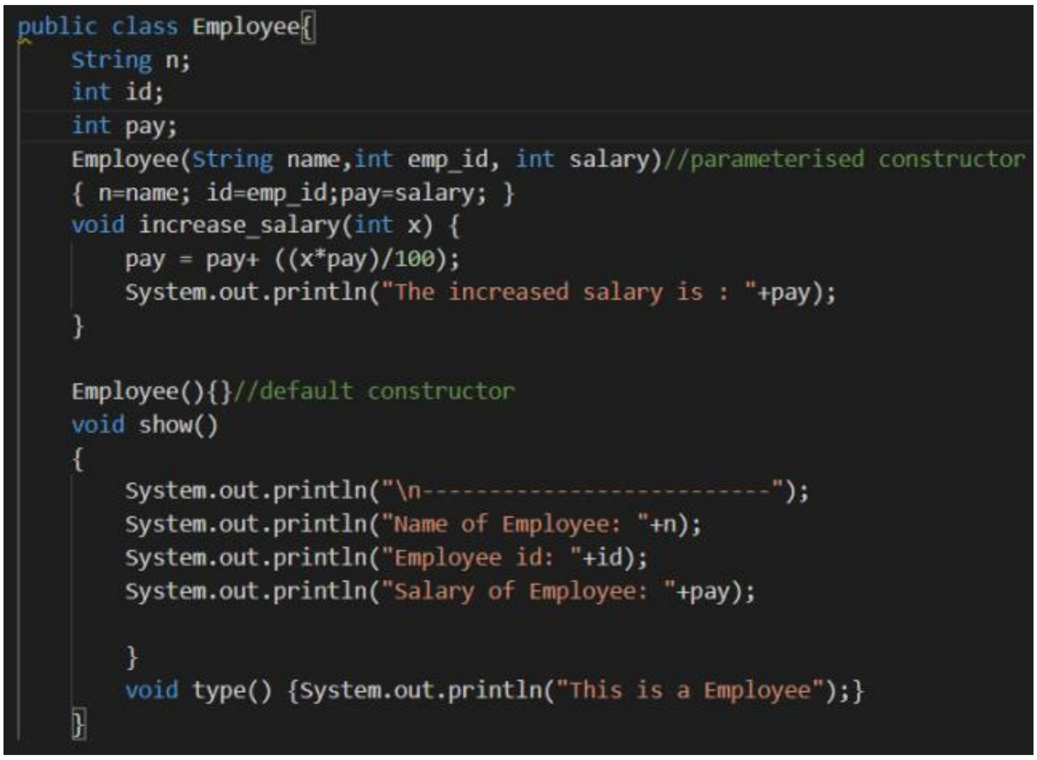
Output



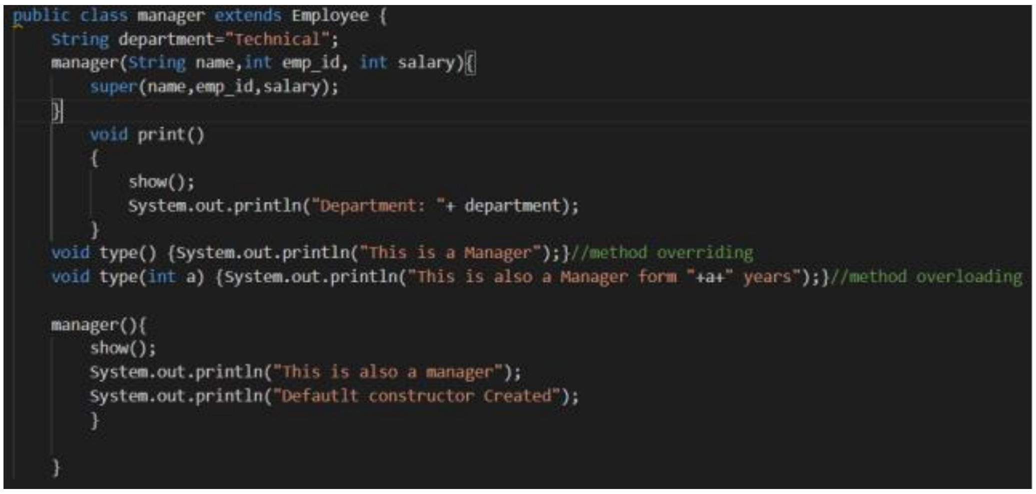
```
Enter call type (Urgent, Lightening, Ordinary))
Urgent
Enter call duration
5
Type of call : Urgent
Call Duration : 5.0
Amount : 20.0
```

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 the employee. Add an instance variable named department to the manager class. Supply a test program that uses these classes and methods.

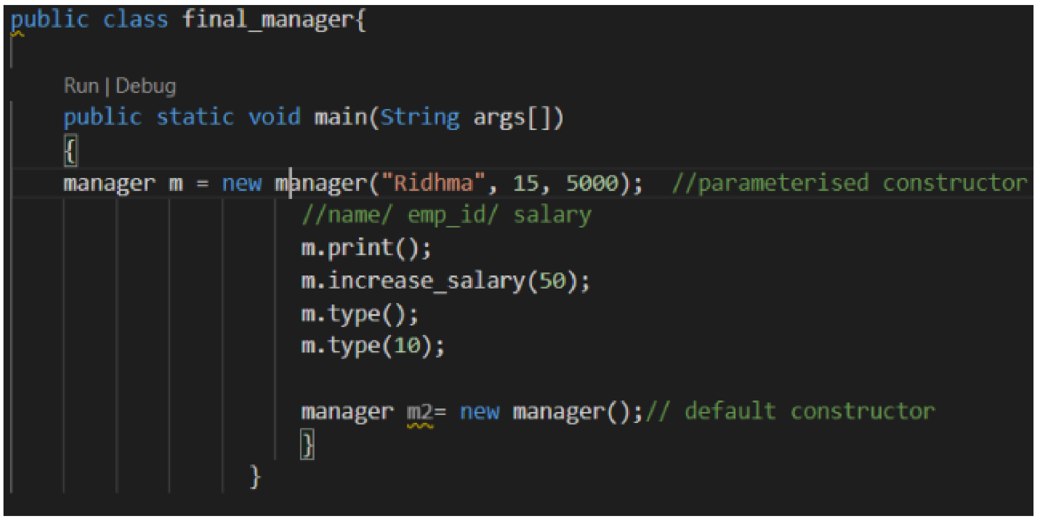
Code



```java
public class Employee{
    String n;
    int id;
    int pay;
    Employee(String name,int emp_id, int salary)//parameterised constructor
    { n=name; id=emp_id;pay=salary; }
    void increase_salary(int x) {
        pay = pay+ ((x*pay)/100);
        System.out.println("The increased salary is : "+pay);
    }

    Employee(){}//default constructor
    void show()
    {
        System.out.println("\n-------------------------");
        System.out.println("Name of Employee: "+n);
        System.out.println("Employee id: "+id);
        System.out.println("Salary of Employee: "+pay);

        }
        void type() {System.out.println("This is a Employee");}
    }
```

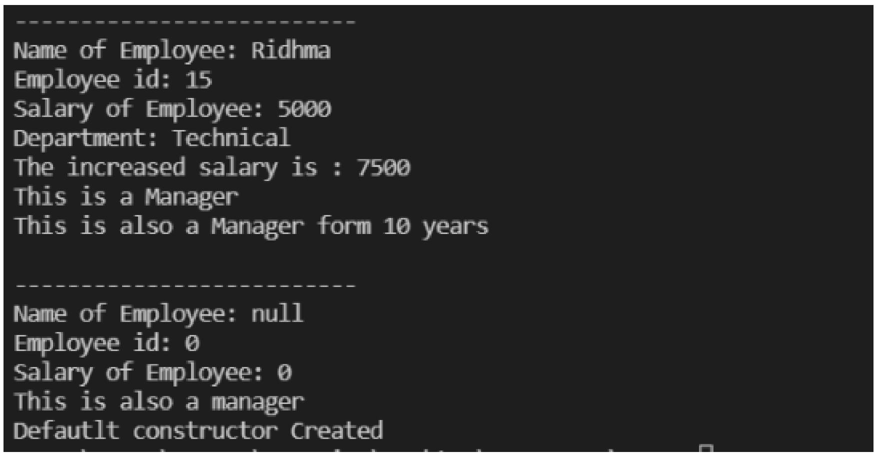


```java
public class manager extends Employee {
    String department="Technical";
    manager(String name,int emp_id, int salary){
        super(name,emp_id,salary);
    }
        void print()
        {
            show();
            System.out.println("Department: "+ department);
        }
    void type() {System.out.println("This is a Manager");}//method overriding
    void type(int a) {System.out.println("This is also a Manager form "+a+" years");}//method overloading

    manager(){
        show();
        System.out.println("This is also a manager");
        System.out.println("Defautlt constructor Created");
        }

    }
```



```
public class final_manager{

    Run | Debug
    public static void main(String args[])
    {
    manager m = new manager("Ridhma", 15, 5000);  //parameterised constructor
                  //name/ emp_id/ salary
                  m.print();
                  m.increase_salary(50);
                  m.type();
                  m.type(10);

                  manager m2= new manager();// default constructor
                  }
              }
```

Output



```
--------------------------
Name of Employee: Ridhma
Employee id: 15
Salary of Employee: 5000
Department: Technical
The increased salary is : 7500
This is a Manager
This is also a Manager form 10 years

--------------------------
Name of Employee: null
Employee id: 0
Salary of Employee: 0
This is also a manager
Defautlt constructor Created
```