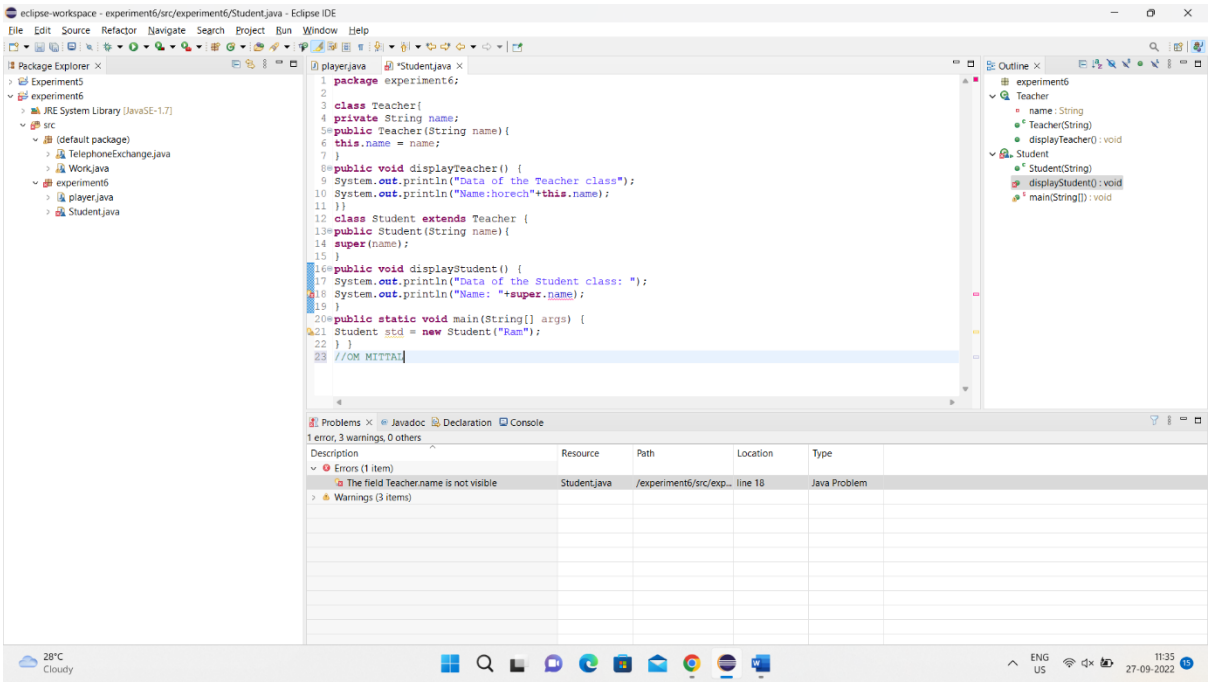
EXPERIMENT 6

1) Write a Java program to show that private member of a super class cannot be accessed from derived classes.

SCREENSHOT:-

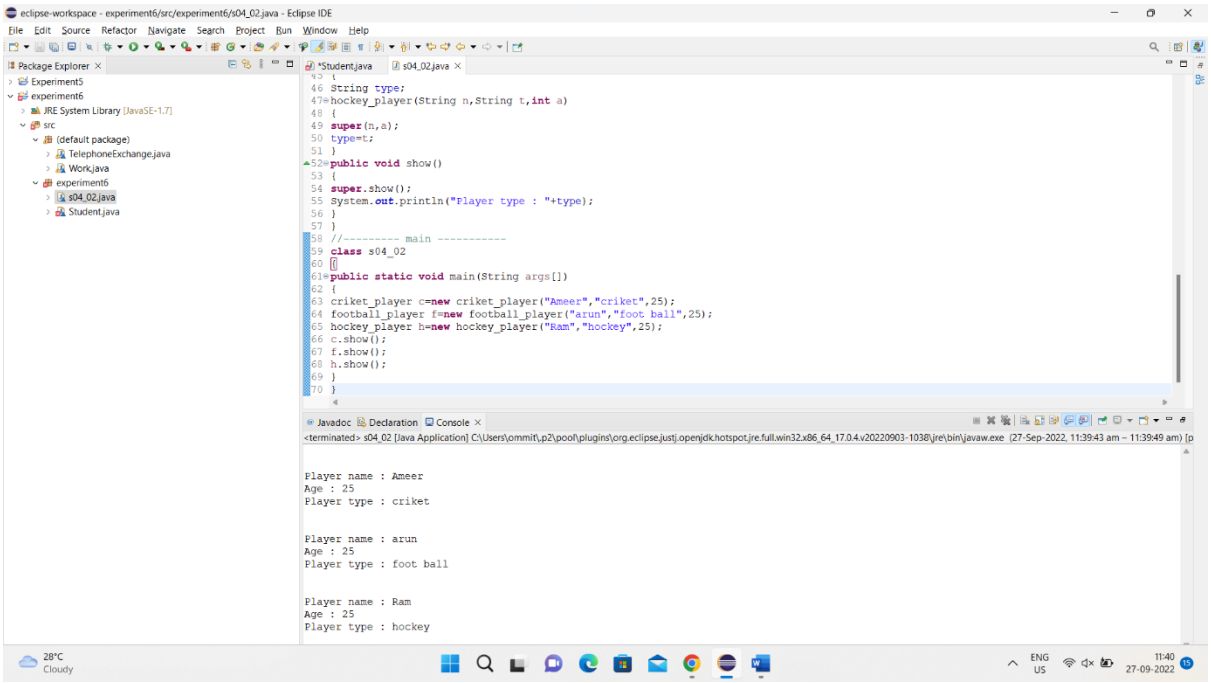


CODE:-

```
package experiment6;


class Teacher{
private String name;
public Teacher(String name){
this.name = name;
}
public void displayTeacher() {
System.out.println("Data of the Teacher class");
System.out.println("Name:horech"+this.name);
}}
class Student extends Teacher {
public Student(String name){
super(name);
}
public void displayStudent() {
System.out.println("Data of the Student class: ");
System.out.println("Name: "+super.name);
}
public static void main(String[] args) {
Student std = new Student("Ram");
} }
//OM MITTAL
```

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

PICTURE:-



CODE:-

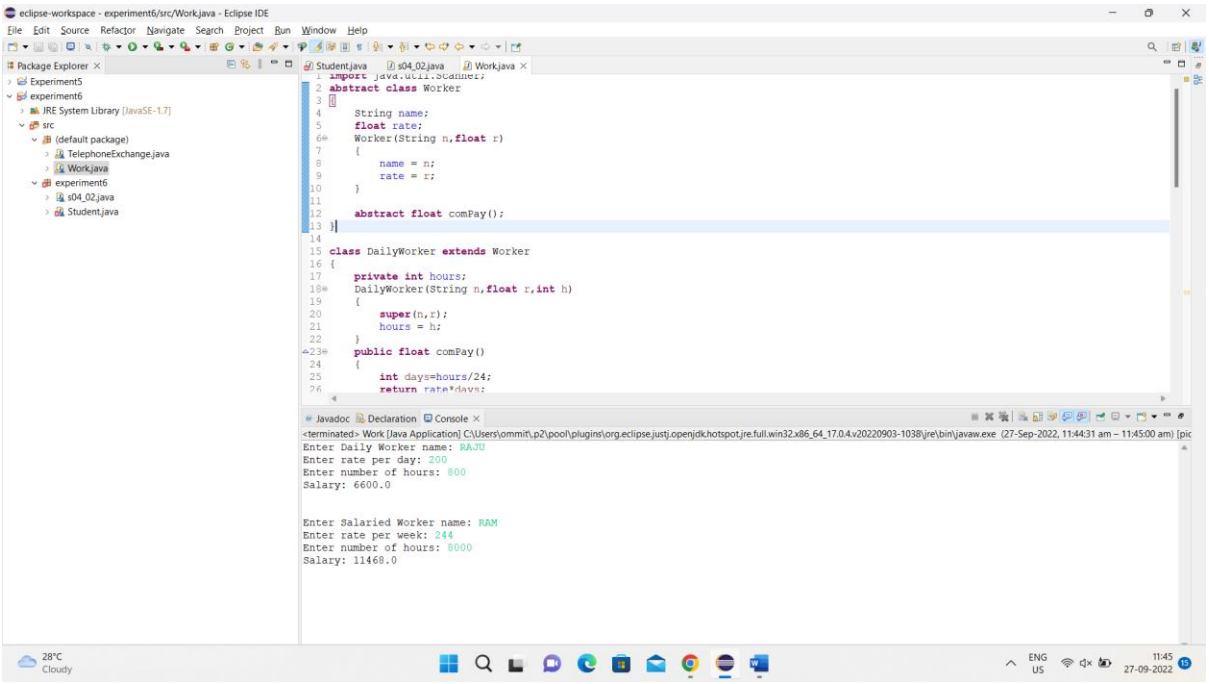
```
package experiment6;


class player
{
String name;
int age;
player(String n,int a)
{ name=n; age=a; }
void show()
{
System.out.println("\n");
System.out.println("Player name : "+name);
System.out.println("Age : "+age);
}
}
class criket_player extends player
{
String type;
criket_player(String n,String t,int a)
{
super(n,a);
type=t;
}
public void show()
{
super.show();
System.out.println("Player type : "+type);
}
}
class football_player extends player
```

```
{
String type;
football_player(String n,String t,int a)
{
super(n,a);
type=t;
}
public void show()
{
super.show();
System.out.println("Player type : "+type);
}
}
class hockey_player extends player
{
String type;
hockey_player(String n,String t,int a)
{
super(n,a);
type=t;
}
public void show()
{
super.show();
System.out.println("Player type : "+type);
}
}
//--------- main -----------
class s04_02
{
public static void main(String args[])
```

```
{
criket_player c=new criket_player("Ameer","criket",25);
football_player f=new football_player("arun","foot ball",25);
hockey_player h=new hockey_player("Ram","hockey",25);
c.show();
f.show();
h.show();
}
}
```

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 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.

SCREENSHOT:-



CODE:-

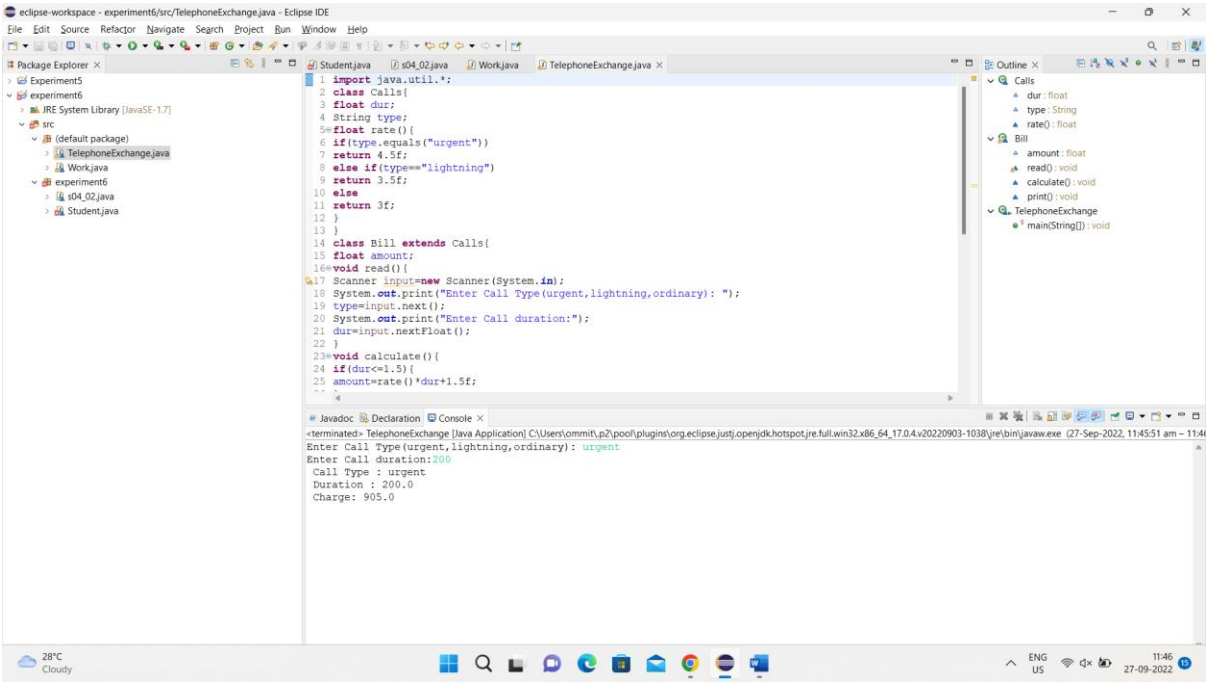
```
import java.util.Scanner;
abstract class Worker
{
   String name;
   float rate;
   Worker(String n,float r)
   {
      name = n;
      rate = r;
   }

   abstract float comPay();
}

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;
   }
}

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 rate*weeks;
   }
}

class Work
{
   public static void main(String args[])
   {
      String name;
      float rate;
      int time;

      Scanner in = new Scanner(System.in);

      System.out.print("Enter Daily Worker name: ");
      name = in.nextLine();
      System.out.print("Enter rate per day: ");
      rate = in.nextFloat();
      System.out.print("Enter number of hours: ");
      time = in.nextInt();
      DailyWorker dw = new DailyWorker(name,rate,time);
```

```
        System.out.println("Salary: "+dw.comPay()+"\n\n");



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

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.

SCREENSHOT:-



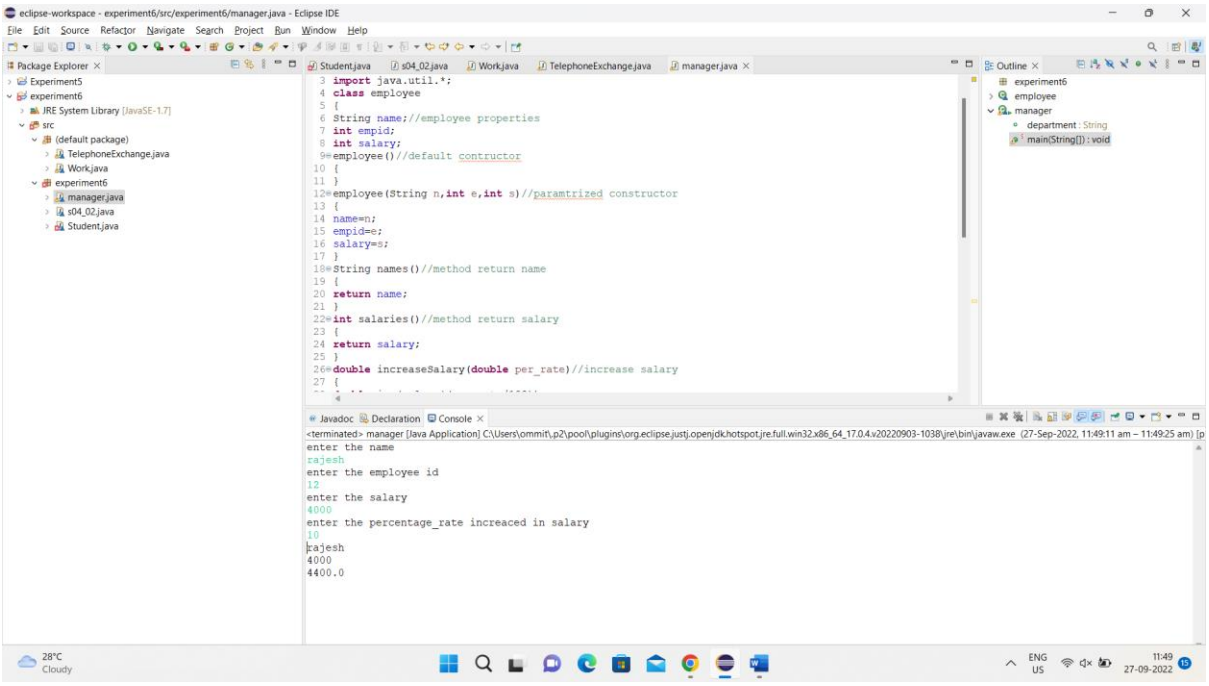
Code:-

```
import java.util.*;
class Calls{
float dur;
String type;
float rate(){
if(type.equals("urgent"))
return 4.5f;
else if(type=="lightning")
return 3.5f;
else
return 3f;
}
}
class Bill extends Calls{
float amount;
void read(){
Scanner input=new Scanner(System.in);
System.out.print("Enter Call Type(urgent,lightning,ordinary): ");
type=input.next();
System.out.print("Enter Call duration:");
dur=input.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(" Call Type : "+type);
System.out.println(" Duration : "+dur);
System.out.println(" Charge: "+amount);
}
}
class TelephoneExchange{
public static void main(String arg[]){
Bill b=new Bill();
b.read();
b.calculate();
b.print();
}
}
```

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 theses classes and methods.

Screenshot:-



Code:-

```
package experiment6;

import java.util.*;

class employee

{

String name;//employee properties

int empid;

int salary;

employee()//default contructor

{

}

employee(String n,int e,int s)//paramtrized constructor

{

name=n;
```

```
empid=e;
salary=s;
}
String names()//method return name
{
return name;
}
int salaries()//method return salary
{
return salary;
}
double increaseSalary(double per_rate)//increase salary
{
double in=(salary*(per_rate/100));
double s=salary+in;
return s;
}
}
class manager extends employee
{
public String department="d";

public static void main(String args[])
{
Scanner sc= new Scanner(System.in);
System.out.println("enter the name");
String n=sc.next();
System.out.println("enter the employee id");
int e=sc.nextInt();
System.out.println("enter the salary");
int s=sc.nextInt();
```

```
System.out.println("enter the percentage_rate increaced in salary");
double p=sc.nextDouble();
employee e1=new employee(n,e,s);
System.out.println(e1.names());
System.out.println(e1.salaries());
System.out.println(e1.increaseSalary(p));
}
}
//om mittal
```