Name: Hitendra Sisodia
Sap id: 500091910

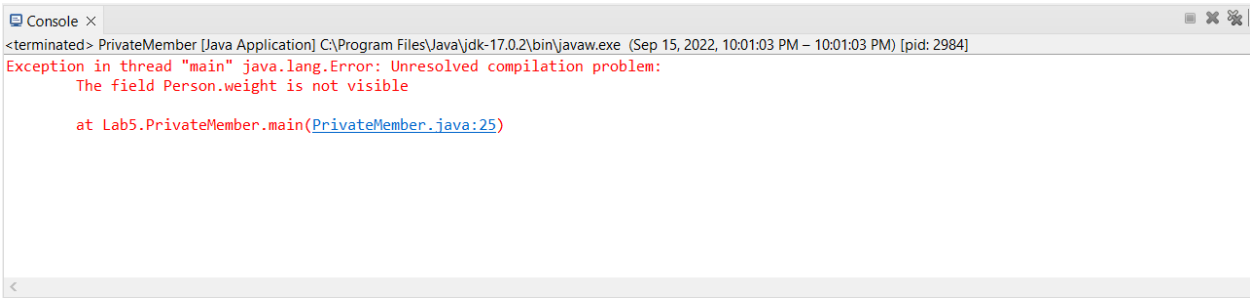
Lab 5

Ques1: Write a Java program to show that private member of a super class cannot be accessed from derived classes.

Source Code

```
package Lab5;

class Person
{
	String name = "Hitendra";
	private float weight = 70;
}
class Student extends Person
{
	Student(){
		System.out.println("Inside student class constructor");
	}
	int studentId = 500091910;
}
public class PrivateMember {
	public static void main(String args[]) {
		System.out.println("Hitendra Sisodia");
		System.out.println("500091910");
		// creation of sub class object
		Student obj = new Student();
		// accessing public members of class
		System.out.println("Name: "+obj.name);
		System.out.println("Student Id: "+obj.studentId);
		// accessing private members of super class
		//System.out.println(obj.weight);
	}
}
```

Output



Name: Hitendra Sisodia
Sap id: 500091910

Lab 5

Ques2: Write a program in Java to create a Player class. Inherit the classes Cricket _Player, Football _Player and Hockey_ Player from Player class.

Source Code

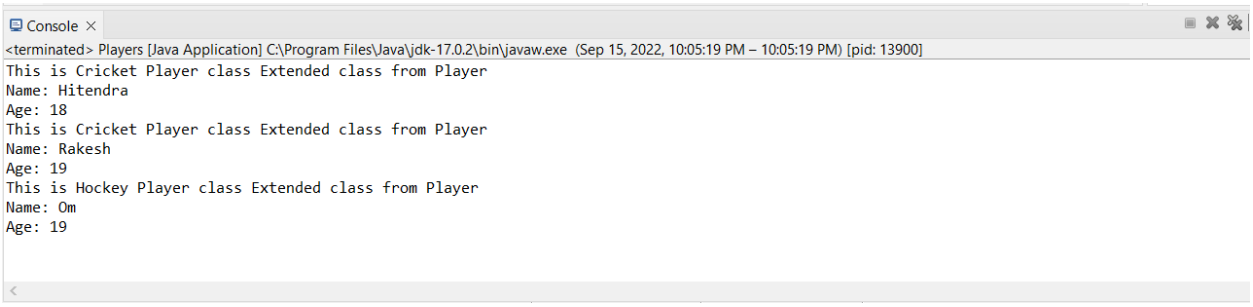
```
package Lab5;

class Player{
	String name;
	int age;
	Player(String name,int age){
		this.name = name;
		this.age = age;
	}
}
class Cricket_Players extends Player{
	Cricket_Players(String name,int age){
		super(name,age);		// Used to trigger parent class
constructor
	}
	void display() {
		System.out.println("This is Cricket Player class Extended class from
Player");
		System.out.println("Name: "+this.name);
		System.out.println("Age: "+this.age);
	}
}
class Football_Players extends Player{
	Football_Players(String name,int age){
		super(name,age);
	}
	void display() {
		System.out.println("This is Cricket Player class Extended class from
Player");
		System.out.println("Name: "+this.name);
		System.out.println("Age: "+this.age);
	}
}
class Hockey_Players extends Player{
	Hockey_Players(String name,int age){
		super(name,age);
	}
	void  display() {
		System.out.println("This is Hockey Player class Extended class from
Player");
		System.out.println("Name: "+this.name);
		System.out.println("Age: "+this.age);
	}
}
public class Players{
	public static void main(String args[]) {
		Cricket_Players cp = new Cricket_Players("Hitendra",18);
		Football_Players fp = new Football_Players("Rakesh",19);
```

Name: Hitendra Sisodia
Sap id: 500091910

Lab 5

```
            Hockey_Players hp = new Hockey_Players("Om",19);
            cp.display();
            fp.display();
            hp.display();
        }
}
```

Output



Name: Hitendra Sisodia
Sap id: 500091910

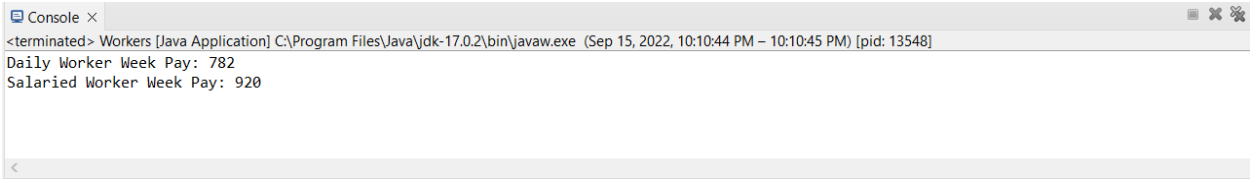
Lab 5

Ques3: 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.

Source Code

```
package Lab5;
class Worker{
	String name;
	int rate = 23;
}
class DailyWorker extends Worker{
	DailyWorker(String name){
		this.name = name;
	}
	void ComPay(int hours) {
		System.out.println("Daily Worker Week Pay: "+rate*hours);
	}
}
class SalariedWorker extends Worker{
	SalariedWorker(String name){
		this.name = name;
	}
	void ComPay(int hours) {
		System.out.println("Salaried Worker Week Pay: "+40*rate);
	}
}
public class Workers{
	public static void main(String args[]) {

		DailyWorker d = new DailyWorker("Hitendra");
		d.ComPay(34);

		SalariedWorker s = new SalariedWorker("Rakesh");
		s.ComPay(34);
	}
}
```

Output



Name: Hitendra Sisodia
Sap id: 500091910

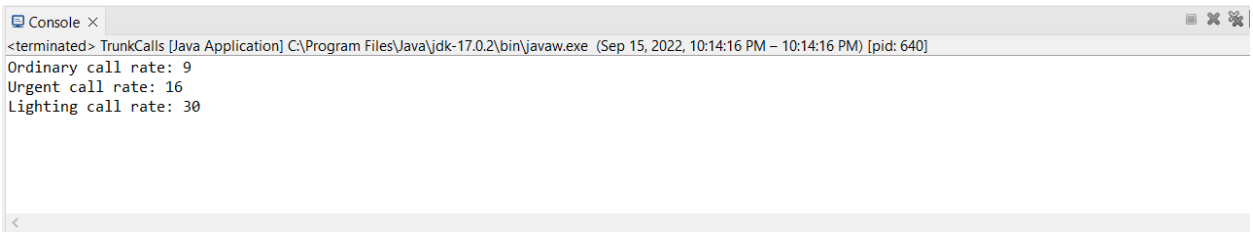
Lab 5

Ques4: 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.

Source Code

```
package Lab5;
class Trunk_call{
	//declare the rate of each call
	int ordinary_rate=3;
	int urgent_rate=4;
	int lighting_rate=5;
}
class ordinary extends Trunk_call{
	//methods
	void charges(int hours){
	System.out.println(this.ordinary_rate*hours);
	}
}
class urgent extends Trunk_call{
	void charges(int hours){
	System.out.println(this.urgent_rate*hours);
	}
}
class lighting extends Trunk_call{
	void charges(int hours){
	System.out.println(this.lighting_rate*hours);
	}
}
class TrunkCalls{
	public static void  main(String args[]){
		ordinary objo =new ordinary();
		objo.charges(3);
		urgent obju=new urgent();
		obju.charges(4);
		lighting objl=new lighting();
		objl.charges(6);
	}
}
```

Output



Name: Hitendra Sisodia
Sap id: 500091910

Lab 5

Ques5: 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.

Source Code

```
package Lab5;
class Employee
{
	String name;
	int empid;
	float salary;
	Employee(){
		System.out.println("This is defulat constructor");
	}
	Employee(String name,int empid,float salary){
		this.name = name;
		this.empid = empid;
		this.salary = salary;
	}
	String get_name() {
		return this.name;
	}
	float get_salary() {
		return this.salary;
	}
	void increaseSalary(int perc) {
		this.salary += (perc/100.0f) * this.salary;
	}
}
class Manager extends Employee{
	String department;
	Manager(String dep){
		this.department = dep;
	}
}
public class Employees {
	public static void main(String args[]) {
		System.out.println("Hitendra Sisodia");
		System.out.println("500091910");
		Employee e1 = new Employee("Hitendra",500091910,1000000f);
		System.out.println("Name: "+e1.get_name());
		System.out.println("Salary: "+e1.get_salary());
		e1.increaseSalary(20);
		System.out.println("Salary: "+e1.get_salary());
		Manager m1 = new Manager("Om");
	}
}
```

Name: Hitendra Sisodia
Sap id: 500091910

Lab 5

Output