**Name =Allahdad**

**Section=D**

**CMS(023-22-0056)**

**Lab 08**

Q1: Create multiple inheritance using interfaces. Create a class Person that derived from class Employee and Officer class. Employee class contains details( ) method and Officer class contains basic info( ) method.You can simply type display any text in details and info methods.

CODE

interface Employee

{

public void details( );

}

interface Officers

{

public void info();

}

class Person implements Employee,Officers

{

public void details()

{

System.out.println("Detail Method of Employee");

}

public void info()

{

System.out.println("Info Method of Officers");

}

}

class Q1

{

public static void main(String args[])

{

Person obj=new Person();

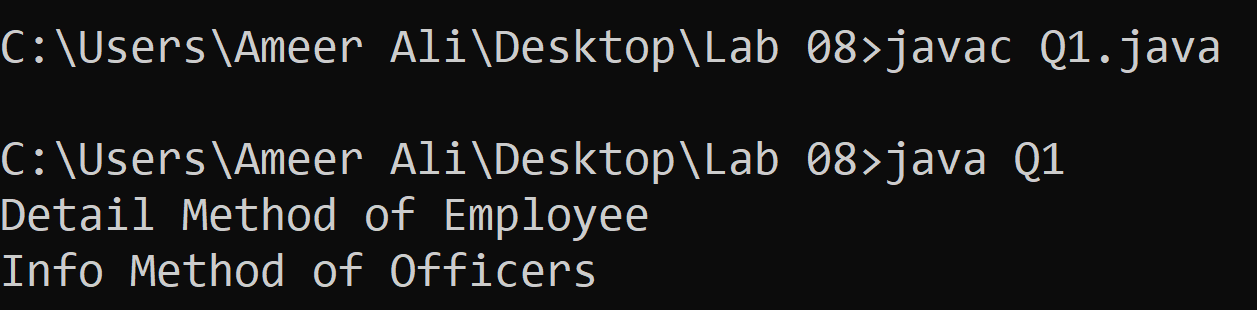
obj.details();

obj.info();

}

}

OUTPUT



Q2:Create three packages namely calculation, departments, and laptops.

1. Put two different classes in each package.
2. Every class should contain any single method.
3. Import all three of these packages in class named PackagePractice, you will have access to 6 classes
4. Call methods of these 6 classes and use them in PackagePractice

CODE

import calculation.\*;

import departments.\*;

import laptops.\*;

public class Main

{

public static void main(String args[])

{

//DEPARTMENTS OBJECT

CSE a=new CSE();

a.cseshow();

BBA b=new BBA();

b.bbashow();

EE c=new EE();

c.eeshow();

System.out.println("");

//CALCULATION OBJECT

Add d=new Add();

d.addshow();

Substract e=new Substract();

e.substractshow();

Multiply f=new Multiply();

f.multiplyshow();

System.out.println("");

//LAPTOPS OBJECT

Acer g=new Acer();

g.acershow();

Dell h=new Dell();

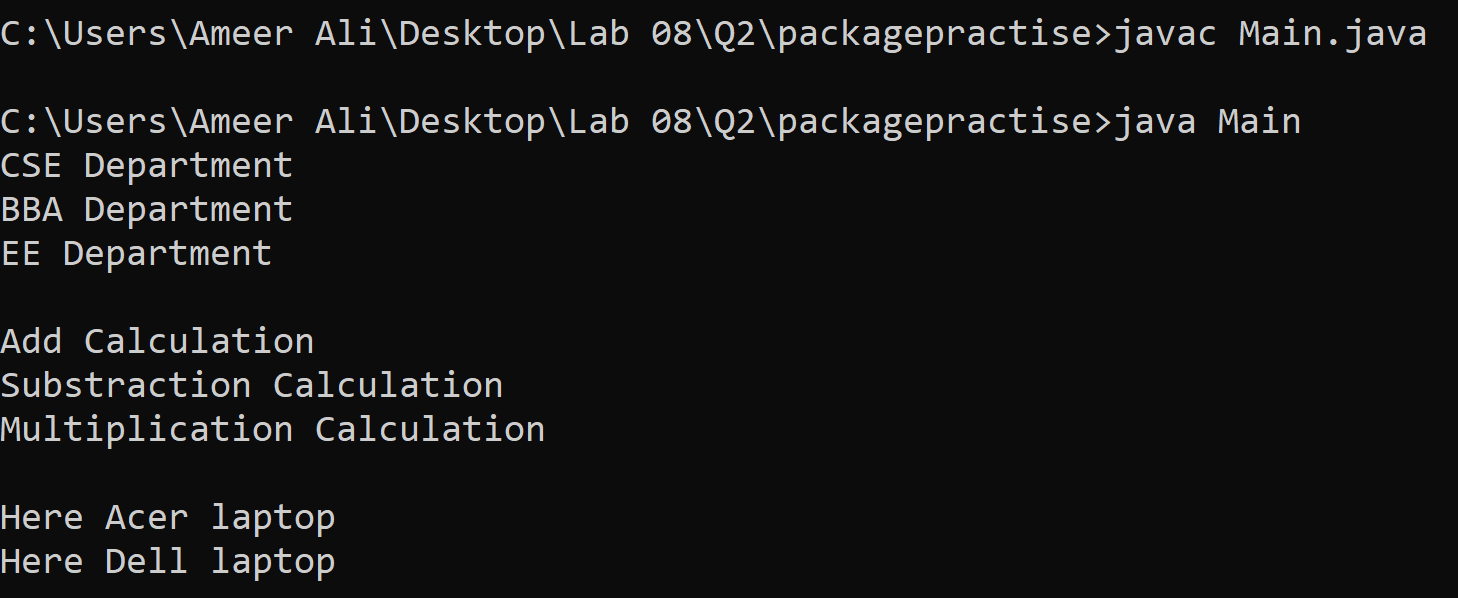
h.dellshow();

System.out.println("");

}

}

OUTPUT



Q3: What is wrong with the following interface?

1.public interface SomethingIsWrong {

void aMethod(int aValue){

System.out.println("Hi Mom");

}

}

Ans:When we make interface or abstract class than atleast one method should be abstract for abstract class and interface without body.

.Not use of public

**Correct Code**

interface SomethingIsWrong {

void aMethod(int aValue);

}

1. Fix the interface in question 1.
2. Is the following interface valid?

public interface Marker {

}

Ans:NO Because use of public.

Q4:

1.Create the Animal interface.

1. Declare abstract method legs.
2. Declare an abstract method eat.

2.Create the Spider, Caterpillar and Cat class that implements animal interface.

1. All classes implement the Animal interface.
2. Implement the eat and legs method.

CODE

interface Animal

{

abstract void leg();

abstract void eat();

}

class Spider implements Animal

{

public void leg()

{

System.out.println("Spider leg method");

}

public void eat()

{

System.out.println("Spider eat method");

}

}

class Caterpillar implements Animal

{

public void leg()

{

System.out.println("Caterpillar leg method");

}

public void eat()

{

System.out.println("Caterpillar eat method");

}

}

class Cat implements Animal

{

public void leg()

{

System.out.println("Cat leg method");

}

public void eat()

{

System.out.println("Cat eat method");

}

}

class Q4

{

public static void main(String args[])

{

Spider spider=new Spider();

spider.leg();

spider.eat();

Caterpillar caterpillar=new Caterpillar();

caterpillar.leg();

caterpillar.eat();

Cat cat=new Cat();

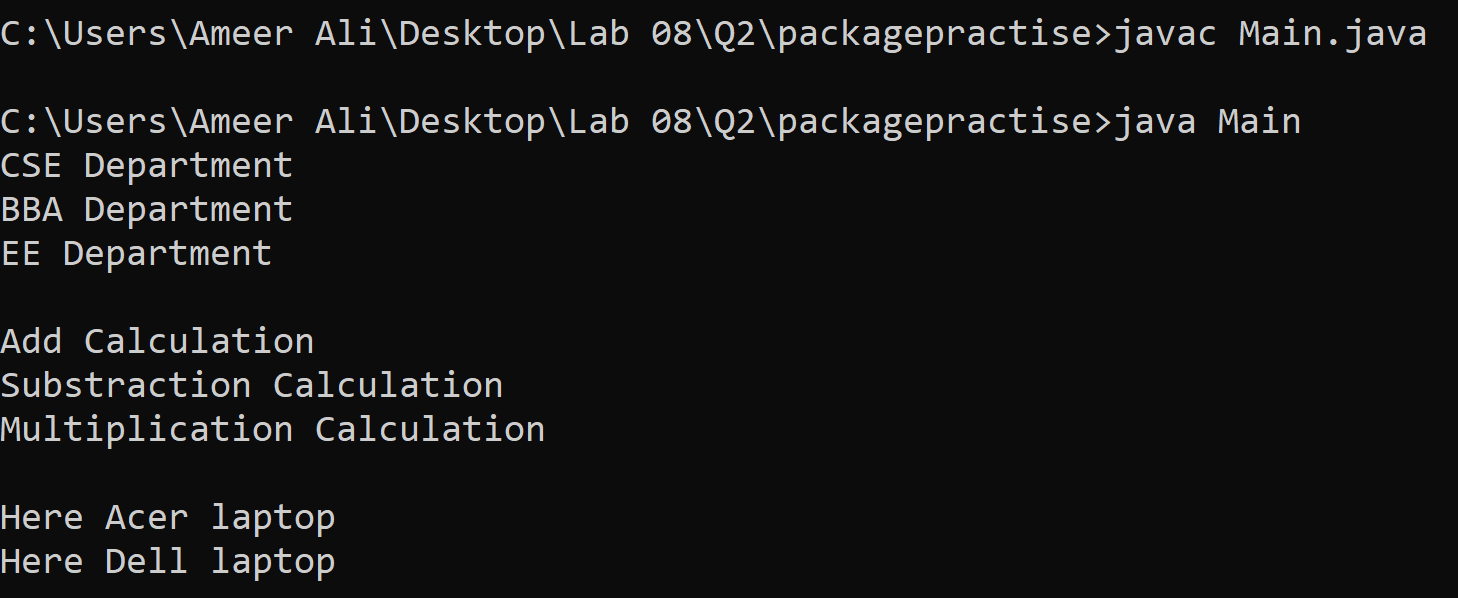
cat.leg();

cat.eat();

}

}

OUTPUT



Q5: We have to calculate the percentage of marks obtained in three subjects (each out of 100) by student A and in four subjects (each out of 100) by student B. Create an abstract class 'Marks' with an abstract method 'getPercentage'. It is inherited by two other classes 'A' and 'B' each having a method with the same name which returns the percentage of the students. The constructor of student A takes the marks in three subjects as its parameters and the marks in four subjects as its parameters for student B. Create an object for each of the two classes and print the percentage of marks for both the students.

CODE

abstract class Marks

{

abstract double getpercentage();

}

class A extends Marks

{

double math,oop,calculus,sum=0,per=0;

A(double math,double oop,double calculus)

{

this.math=math;

this.oop=oop;

this.calculus=calculus;

}

double getpercentage()

{

sum=math+oop+calculus;

per=(sum/300)\*100;

return per;

}

}

class B extends Marks

{

double math,oop,calculus,discrete,sum=0,per=0;

B(double math,double oop,double calculus,double discrete)

{

this.math=math;

this.oop=oop;

this.calculus=calculus;

this.discrete=discrete;

}

double getpercentage()

{

sum=math+oop+calculus+discrete;

per=(sum/400)\*100;

return per;

}

}

class Q5

{

public static void main(String args[])

{

//OBJECT OF A

A a;

a=new A(89.5,76.3,87.00);

System.out.println("The percentage of A Student is "+a.getpercentage()+"%");

//OBJECT OF B

B b;

b=new B(89.5,76.3,87.1,87.00);

System.out.println("The percentage of B Student is "+b.getpercentage()+"%");

}

}

OUTPUT

