1) Interface Example - 1

```
interface testInterface {
   final int a = 10;
   void display(); }
class TestClass implements testInterface {
public void display(){
     System.out.println("Geek");  }
}
class Geeks {
   public static void main(String[] args) {
       TestClass t = new TestClass();
       t.display();
       System.out.println(t.a);  }
}
```

OUTPUT:-

```
Geek
10
```

2) Interface Example - 2

```
interface Add{
   int add(int a,int b); }
interface Sub{
   int sub(int a,int b); }
class Cal implements Add , Sub{
   public int add(int a,int b){
       return a+b; }
   public int sub(int a,int b){
       return a-b;  }
}
class GFG{
   public static void main (String[] args) {
       Cal x = new Cal(); System.out.println("Addition : " + x.add(2,1));
       System.out.println("Substraction : " + x.sub(2,1)); }
}
```

OUTPUT:-

```
Addition : 3
Substraction : 1
```

3) Interface Example - 3

```
interface InternalExam {
    int internalMarks1 = 25;
    int internalMarks2 = 28;
    int internalMarks3 = 27;
    void showInternalMarks() }
class ExternalExam {
    int externalMarks1 = 50;
    int externalMarks2 = 48;
    int externalMarks3 = 49;
    void showExternalMarks() {
        System.out.println("External Exam Marks:");
        System.out.println("Subject 1: " + externalMarks1);
        System.out.println("Subject 2: " + externalMarks2);
        System.out.println("Subject 3: " + externalMarks3);  }
}
```

```java
class Result extends ExternalExam implements InternalExam {
    public void showInternalMarks() {
        System.out.println("Internal Exam Marks:");
        System.out.println("Subject 1: " + internalMarks1);
        System.out.println("Subject 2: " + internalMarks2);
        System.out.println("Subject 3: " + internalMarks3);}
    public void displayTotalMarks() {
        int totalInternal = internalMarks1 + internalMarks2 + internalMarks3;
        int totalExternal = externalMarks1 + externalMarks2 + externalMarks3;
        int total = totalInternal + totalExternal;
        System.out.println("\nTotal Internal Marks: " + totalInternal);
        System.out.println("Total External Marks: " + totalExternal);
        System.out.println("Aggregate Total Marks: " + total); }
}
public class ExamResultDemo {
    public static void main(String[] args) {
        Result student = new Result();
        student.showInternalMarks();
        student.showExternalMarks();
        student.displayTotalMarks(); }
}
```

```
Internal Exam Marks:
Subject 1: 25
Subject 2: 28
Subject 3: 27
External Exam Marks:
```

```
Subject 1: 50

Subject 2: 48

Subject 3: 49

Total Internal Marks: 80

Total External Marks: 147

Aggregate Total Marks: 227
```

4) write a java progrm intrface program create 2 interface unit test 1 marks,unit test 2 marks and class is assignment marks totle marks is inherit

```
interface UnitTest1 {

    int getUnitTest1Marks();}

interface UnitTest2 {

    int getUnitTest2Marks(); }

class Assignment implements UnitTest1, UnitTest2 {

    private int unitTest1Marks;

    private int unitTest2Marks;

    private int assignmentMarks;

    public Assignment(int ut1, int ut2, int assignment) {

        this.unitTest1Marks = ut1;

        this.unitTest2Marks = ut2;

        this.assignmentMarks = assignment;}

    public int getUnitTest1Marks() {

        return unitTest1Marks;}

    public int getUnitTest2Marks() {

        return unitTest2Marks;}

    public int getAssignmentMarks() {

        return assignmentMarks; }
    public int getTotalMarks() {
        return unitTest1Marks + unitTest2Marks + assignmentMarks;}
    public void displayMarks() {
        System.out.println("Unit Test 1 Marks: " + getUnitTest1Marks());
        System.out.println("Unit Test 2 Marks: " + getUnitTest2Marks());
        System.out.println("Assignment Marks: " + getAssignmentMarks());
        System.out.println("Total Marks: " + getTotalMarks());}
}
public class Main {
    public static void main(String[] args) {
        Assignment student = new Assignment(25, 30, 40);
        student.displayMarks();}
}
```

```
Unit Test 1 Marks: 25
Unit Test 2 Marks: 30
Assignment Marks: 40
Total Marks: 95
```

THERED:-

1) Thread Example

create 3 thread t1,t2,t3   t1 print roll no t2 print college name t3 print student nam,e set thread priority such a way that it will print

out on following way college roll no student name

```
class CollegeThread extends Thread {
    public void run() {
        System.out.println("College Name: ABC College");}
}
```

```
class RollNoThread extends Thread {

   public void run() {

      System.out.println("Roll No: 12345");}

}

class StudentNameThread extends Thread {

   public void run() {

      System.out.println("Student Name: John Doe");}

}

public class main {

   public static void main(String[] args) {

      CollegeThread t2 = new CollegeThread();

      RollNoThread t1 = new RollNoThread();

 StudentNameThread t3 = new StudentNameThread();
t2.setPriority(Thread.MAX_PRIORITY);  t1.setPriority(Thread.NORM_PRIORITY);

t3.setPriority(Thread.MIN_PRIORITY);

   t2.start();

      t1.start();

      t3.start(); }

}
```

```
College Name: ABC College

Student Name: John Doe

Roll No: 12345
```

APPLET:-

1)pass x and y value to the applet and display its addition,subtraction and multiplication value

```
import java.applet.Applet;

import java.awt.*;
```

```
/*
<applet code="SimpleCalcApplet" width=300 height=200>
    <param name="x" value="20">
    <param name="y" value="10">
</applet>
*/
public class SimpleCalcApplet extends Applet {
    int x, y;
    public void init() {
        x=Integer.parseInt(getParameter("x"));       y=Integer.parseInt(getParameter("y")); }
public void paint(Graphics g) {
g.drawString("X = " + x + ", Y = " + y, 20, 20);
        g.drawString("Addition: " + (x + y), 20, 50);
        g.drawString("Subtraction: " + (x - y), 20, 70);
        g.drawString("Multiplication: " + (x * y), 20, 90); }
}
```

2) pass length,width and reduis is to parameter the applet and display area of rectangle,parameter of squere and area of circle

```
import java.applet.Applet;
import java.awt.*;
/*
<applet code="ShapeApplet" width=300 height=200>
    <param name="length" value="10">
    <param name="width" value="5">
    <param name="radius" value="7">
    <param name="side" value="4">
```

```
</applet>
*/
public class ShapeApplet extends Applet {
    int length, width, radius, side;
    public void init() {
        length=Integer.parseInt(getParameter("length"));
width=Integer.parseInt(getParameter("width"));
radius=Integer.parseInt(getParameter("radius"));
side=Integer.parseInt(getParameter("side")); }
    public void paint(Graphics g) {
  double areaRectangle = length * width;
double areaCircle=3.14 * radius * radius;
        double perimeterSquare = 4 * side;
        g.drawString("Area of Rectangle: " + areaRectangle, 20, 20);
        g.drawString("Area of Circle: " + areaCircle, 20, 40);
        g.drawString("Perimeter of Square: " + perimeterSquare, 20, 60);}
}
```

3) pass principal amount no of year and reate enter as a paremeter to applet and display simple inheritance

```
import java.applet.Applet;
import java.awt.*;
/*
<applet code="SimpleInterestApplet" width=300 height=150>
    <param name="principal" value="1000">
    <param name="rate" value="5">
    <param name="time" value="2">
</applet>
```

```
*/

public class SimpleInterestApplet extends Applet {

   double principal, rate, time, interest;

   public void init() {

principal=Double.parseDouble(getParameter("principal"))

rate=Double.parseDouble(getParameter("rate"));
time=Double.parseDouble(getParameter("time"));

 interest = (principal * rate * time) / 100}

   public void paint(Graphics g) {

g.drawString("Principal: " + principal, 20, 20);

      g.drawString("Rate: " + rate + "%", 20, 40);

      g.drawString("Time: " + time + " years", 20, 60);

      g.drawString("Simple Interest: " + interest, 20, 80)}

}
```