1. **Write a program to create a class Student2 along with two method getData(), printData() to get the value through argument and display the data in printData. Create the two objects s1 ,s2 to declare and access the values from class STtest.**

```
class Student2
{
String name;
int age;
String city;

void getData(String name, int age, String city)
{
this.name = name;
this.age = age;
this.city = city;
}

void printData()
{
System.out.println("Name: "+name);
System.out.println("Age : "+age);
System.out.println("City: "+city);
}
}

class a2prog2
{
public static void main(String args[])
{
Student2 s1 = new Student2();
Student2 s2 = new Student2();
s1.getData("Karan",21,"Mangalore");
s2.getData("Kavitha",21,"Udupi");
s1.printData();
s2.printData();
}
}
```

2. **Write a Java program to calculate total, average and grade for 10 students. Consider the necessary datamembers and member methods required to achieve the requirements**

```
import java.util.Scanner;

class student1 {
    int mark[] = new int[3];
    int total = 0;
    float avg;
    char grade;

    student1() {
        Scanner sc = new Scanner(System.in);

        for (int i = 0; i < 3; i++) {
            do {
                System.out.println("marks in sub " + (i + 1) + ":");
                mark[i] = sc.nextInt();
            } while (mark[i] < 0 || mark[i] > 100);
        }
    }

    void calgrade() {
        for (int i = 0; i < 3; i++) {
            total += mark[i];
        }

        avg = (float) total / 3;

        for (int i = 0; i < 3; i++) {
            if (mark[i] <= 35) {
                grade = 'F';
                return;
```

```
            return;
        }

        if (avg <= 35)
            grade = 'F';

        else if (avg <= 50)
            grade = 'C';

        else if (avg <= 65)
            grade = 'B';

        else if (avg <= 80)
            grade = 'A';

        else if (avg <= 90)
            grade = 'E';

        else
            grade = 'O';
    }
}

void setresult() {
    for (int i = 0; i < 3; i++) {
        System.out.println("Mark " + (i + 1) + ":" + mark[i]);
    }
    System.out.println("Total :" + total);
    System.out.println("Grade :" + grade);
    System.out.println("Average :" + avg);
```

```
    System.out.println("Average :" + avg);
    System.out.println(x:"");
}
}

public class result {
    Run | Debug
    public static void main(String[] args) {
        student1 s[] = new student1[10];

        for (int i = 0; i < 3; i++) {
            System.out.println("Student " + (i + 1));
            s[i] = new student1();
            s[i].calgrade();
            s[i].setresult();
        }
    }
}
```

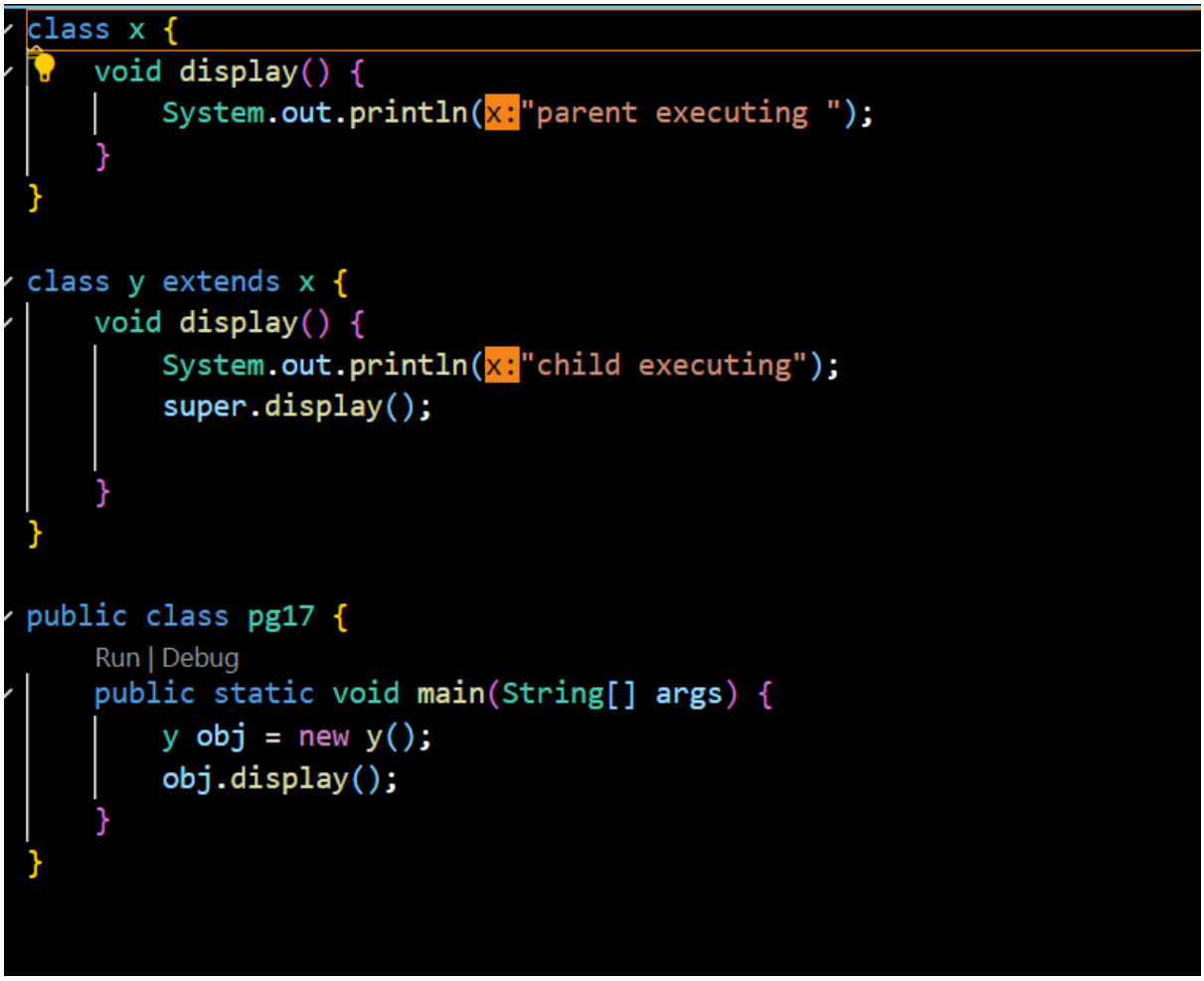
Construct overload

```
class sum {
    int s;

    sum(int a, int b) {
        s = a + b;
        System.out.println("sum :" + s);
    }

    sum(int a, int b, int c)

    {
        s = a + b + c;
        System.out.println("sum :" + s);
    }

}

public class q16 {
    public static void main(String[] args) {
        sum k = new sum(10, 20);
        sum k1 = new sum(10, 20, 30);
    }
}
```

Method overload(compile time polymorphism)

```java
class sum1 {
    int s;

    void sum1(int a, int b) {
        s = a + b;
        System.out.println("sum :" + s);
    }

    void sum1(int a, int b, int c)

    {
        s = a + b + c;
        System.out.println("sum " + s);
    }
}

public class q17 {
    public static void main(String[] args) {
        sum1 w = new sum1();
        w.sum1(20, 45);
        w.sum1(10, 20);
    }
}
```

Method override (run time polymorphism)



```
class x {
    void display() {
        System.out.println(x:"parent executing ");
    }
}

class y extends x {
    void display() {
        System.out.println(x:"child executing");
        super.display();

    }
}

public class pg17 {
    Run | Debug
    public static void main(String[] args) {
        y obj = new y();
        obj.display();
    }
}
```

Upcasting

If the reference variable of Parent class refers to the object of Child class, it is known as upcasting.

```
class A{}
class B extends A{}

A a=new B();//upcasting
```

3. **Write a program to demonstrate upcasting and compile time polymorphism**

```
class over {
    void meth1(int rollno, String name, int marks) {
        System.out.println("Roll no :" + rollno + " Name :" + name + " marks :" + marks);
    }

    void meth2(int empno, String City) {
        System.out.println("Empno :" + empno + " City :" + City);
    }

    void meth2(int empno, String City, int salary) {
        System.out.println("Empno :" + empno + " City :" + City + " Salary :" + salary);
    }
}

class upcast extends over {
    void meth1(int rollno, String name, int marks) {
        System.out.println("Roll no :" + rollno + " Name :" + name + " marks :" + marks);
    }
}

class upcastover {
    public static void main(String[] args) {
        over s1 = new over();
        over s2 = new upcast();

        s1.meth1(101, "Teena", 85);
        s1.meth2(1, "Udupi");
        s1.meth2(4, "manglore", 12000);
        s2.meth1(102, "heena", 56);
    }
}
```

4. **Write a program to create a class named shape. In this class we have three sub classes circle, triangle and square each class has two member function named draw () and erase (). Create these using polymorphism concepts.**

```
class shape {
    void draw() {
        System.out.println("Drawing a Shape");
    }

    void erase() {
        System.out.println("Erasing a shape");
    }
}

class circle extends shape {
    void draw() {
        System.out.println("Drawing a circle");
    }

    void erase() {
        System.out.println("Erasing a cirlce");
    }
}

class triangle extends shape {
    void draw() {
        System.out.println("Drawing a triangle");
    }

    void erase() {
        System.out.println("Erasing a triangle");
    }
}

class square extends shape {
    void draw() {
        System.out.println("Drawing a square");
    }
```

```java
class square extends shape {
    void draw() {
        System.out.println(x:"Drawing a square");
    }

    void erase() {
        System.out.println(x:"Erasing a square");
    }
}

public class main {
    Run | Debug
    public static void main(String[] args) {
        shape s1 = new circle();
        shape s2 = new triangle();
        shape s3 = new square();

        s1.draw();
        s1.erase();

        s2.draw();
        s2.erase();

        s3.draw();
        s3.erase();
    }
}
```

5. Overload area() method to calculate the area of three different shapes

```java
public class areacal {
    int answer;

    void area(int l, int b) {
        answer = l * b;
        System.out.println("Area of Rectangle :" + answer);

    }

    void area(double r) {
        double ans = 3.14 * r * r;
        System.out.println("Area of circle :" + ans);

    }

    void area(int b, int h, int l) {
        answer = l * b * h;
        System.out.println("Area of cube :" + answer);

    }

    public static void main(String[] args) {
        areacal cal = new areacal();
        cal.area(2, 3);
        cal.area(2);
        cal.area(2, 3);

    }
}
```

6. Write a java program to demonstrate use of pakages and implement interface inheritance.

Step1: Create a folder as Student

Step2: Create a file Student.java in created folder.

```
package student;
public interface student
{
public void Rollno();
public void Course();
public void marks();
}
```

Step3:Create a file main1.java outside the student folder

```
import student.*;

class test implements students {
    public void rollno() {
        System.out.println("g100");
    }

    public void name() {
        System.out.println("heena");
    }

    public void course() {
        System.out.println("mca");
    }
}

public class main1 {
    public static void main(String[] args) {
        test t1 = new test();
        t1.rollno();
        t1.name();
        t1.course();
    }
}
```

7. Write a program to demonstrate difference between abstract class and interface

```java
abstract class adder {
    void add(int a, int b) {
        System.out.println("addition :" + (a + b));
    }

    abstract void mul(int a, int b);

}
class multi extends adder {
    void mul(int a, int b) {
        System.out.println("multiplication :" + (a * b));
    }
}

interface subtraction {
    void sub(int a, int b);
}

class subt implements subtraction {
    public void sub(int a, int b) {
        System.out.println("subtraction :" + (a - b));
    }
}

public class test {
    Run | Debug
    public static void main(String[] args) {
        adder a = new multi();
        subtraction b = new subt();
        a.add(a:6, b:2);
        a.mul(a:6, b:2);
        b.sub(a:6, b:2);
    }
}
```

8. **Write a Java program to count the number of characters in a file.**

```java
import java.io.*;

public class filecount {
    public static void main(String[] args) throws IOException {
        FileInputStream fis = new FileInputStream("C:\\Users\\Priyanka\\
        int ch, count = 0;

        try {
            while ((ch = fis.read()) != -1) {
                count++;
            }
            System.out.println(count);

        }

        catch (Exception e) {
        }
    }

}
```

9. **Copy file**

```java
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

class copyfile {
    public static void main(String[] args) throws IOException {
        FileInputStream r = new FileInputStream("C:\\Users\\Priya
        FileOutputStream w = new FileOutputStream("C:\\Users\\Pri
        int i;
        while ((i = r.read()) != -1) {
            w.write((char) i);
        }
        System.out.println("Data copied successfully");
    }
}
```

```
ws IOException {
("C:\\Users\\Priyanka\\Desktop\\java\\student\\file1.txt.txt");
am("C:\\Users\\Priyanka\\Desktop\\java\\student\\copy.txt.txt");
```

10 . writing Byte array to file

```
import java.io.*;

public class write {
    public static void main(String[] args) {

        try {
            byte buffer[] = new byte[80];
            String msg = "hello hii how are you blah ";
            buffer = msg.getBytes();
            FileOutputStream fos = new FileOutputStream("C:\\Users\\Priyanka

            fos.write(buffer);
            fos.flush();
            fos.close();
        }

        catch (FileNotFoundException e) {
            System.out.println(e);
        } catch (IOException e) {
            System.out.println(e);
        }
    }
}
```

10. Create Thread

```java
class mythread extends Thread {
    public void run() {
        System.out.println("thread is running");
    }
}

class cthread {
    public static void main(String[] args) {
        mythread mt = new mythread();
        mt.start();
    }
}
```

11. Display current Thread

```java
class curthread {
    public static void main(String[] args) {

        Thread t = Thread.currentThread();
        System.out.println("Current Thread :" + t);
        t.setName("MY Thread");
        System.out.println("thread name :" + t.getName());
        System.out.println("priority :" + t.getPriority());

        try {
            for (int i = 0; i < 5; i++) {
                System.out.println(i);
                Thread.sleep(1000);
            }
        } catch (InterruptedException e) {
            System.out.println("Thread is interupted ..");
        }
    }
}
```

12. Multi thread

```java
class mythread extends Thread {
    String name;

    mythread(String name) {
        this.name = name;
    }

    public void run() {
        for (int i = 1; i <= 5; i++) {
            System.out.println(name + "count :" + i);

            try {
                Thread.sleep(2000);
            } catch (InterruptedException e) {
                System.out.println("Thread interuppted ");
            }
        }
    }
}

public class multithread {
    public static void main(String[] args) {
        Thread t1 = new mythread("hello thread");
        Thread t2 = new mythread("namaste Thread");
        Thread t3 = new mythread("Tata Thread");

        t1.start();
        t2.start();
        t3.start();
    }
}
```

13. Priority thread

```java
class mythread extends Thread {
    String name;

    mythread(String name) {
        this.name = name;
    }

    public void run() {
        for (int i = 1; i <= 5; i++) {
            System.out.println(name + "count :" + i);

            try {
                Thread.sleep(2000);
            } catch (InterruptedException e) {
                System.out.println("Thread interuppted ");
            }
        }
    }
}

public class priority {
    public static void main(String[] args) {
        Thread t1 = new mythread("lower priority");
        Thread t2 = new mythread("normal priority");
        Thread t3 = new mythread("higher priority");

        t1.setPriority(Thread.MIN_PRIORITY);
        t2.setPriority(Thread.NORM_PRIORITY);
        t3.setPriority(Thread.MAX_PRIORITY);

        t1.start();
        t2.start();
        t3.start();
    }
}
```

14. Synchronized Thread

```
class first {
    synchronized public void display(String msg) {
        System.out.print("[" + msg);
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            System.out.print("Interupted ");
        }
        System.out.println("]");
    }
}

class second extends Thread {
    String msg;
    first fobj;

    second(first fp, String str) {
        fobj = fp;
        msg = str;
        start();
    }

    public void run() {
        fobj.display(msg);
    }
}

public class synchro {
    public static void main(String[] args) {
        first fnew = new first();
        second s1 = new second(fnew, "hello");
        second s2 = new second(fnew, "good evng");
    }
}
```

15. User defined Exception(age)

```java
public class ageExcep {
    public static void main(String[] args) {
        try {
            ValidateAge(17);
            System.out.println("Age is valid");

        } catch (Exception e) {
            System.out.println("Exception caught :" + e.getMessage());
        }
    }

    public static void ValidateAge(int age) throws Exception {
        if (age < 18)
            throw new Exception("Age should be above 18");
    }
}
```

16. Multi catch exception

```java
public class multicatch {
    public static void main(String[] args) {
        try {
            int number[] = { 1, 2, 3 };
            System.out.println(number[4]);
            int result = 10 / 0;
            System.out.println("Result :" + result);
        } catch (ArithmeticException e) {
            System.out.println("airthmetic exception caught :" + e.getMessage());
        }

        catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("array index out  exception caught :" + e.getMessage());
        } catch (Exception e) {
            System.out.println(" exception caught :" + e.getMessage());
        }
    }
}
```

APPLET-FRAME-SWING

1. GCD-Applet

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;

/*<applet code=GcdA width=200 height=200></applet>*/

public class GcdA extends Applet implements ActionListener{
  Label l1,l2,l3;
  TextField t1,t2,t3;
  Button b1,b2;


 public void init(){
   l1=new Label("Num 1");
   l2=new Label("Num 2");
   l3=new Label("answer");
   t1=new TextField();
   t2=new TextField();
   t3=new TextField();
   b1= new Button("Calculate");
   b2=new Button("Clear");
   b1.addActionListener(this);
   b2.addActionListener(this);

   setLayout(new GridLayout(4,2));

   add(l1);
   add(t1);
   add(l2);
   add(t2);
   add(l3);
   add(t3);
   add(b1);
   add(b2);

 }

 public void actionPerformed(ActionEvent ae)
 {
   if(ae.getSource()==b1)
   {
         int a=Integer.parseInt(t1.getText());
         int b=Integer.parseInt(t2.getText());
```

```
        while(a!=b)
        {
              if(a>b)
                    a=a-b;
              else
                    b=b-a;
        }
        t3.setText("GCD :"+a);
     }
   if(ae.getSource()==b2)
   {
        t1.setText("");
        t2.setText("");
        t3.setText("");
        t1.requestFocus();
   }
 }
}
```

---

2. **GCD-Swing**

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

/*<applet code=GcdS width=200 height=200></applet>*/

public class GcdS extends JApplet implements ActionListener{
     JLabel l1,l2,l3;
     JTextField t1,t2,t3;
     JButton b1,b2;


 public void init(){
      l1=new JLabel("Num 1");
      l2=new JLabel("Num 2");
      l3=new JLabel("answer");
      t1=new JTextField();
      t2=new JTextField();
      t3=new JTextField();
      b1= new JButton("Calculate");
      b2=new JButton("Clear");
```

```
        b1.addActionListener(this);
        b2.addActionListener(this);

        Container cp=getContentPane();
        cp.setLayout(new GridLayout(4,2));

        add(l1);
        add(t1);
        add(l2);
        add(t2);
        add(l3);
        add(t3);
        add(b1);
        add(b2);

    }

    public void actionPerformed(ActionEvent ae)
    {
        if(ae.getSource()==b1)
        {
              int a=Integer.parseInt(t1.getText());
              int b=Integer.parseInt(t2.getText());

              while(a!=b)
              {
                    if(a>b)
                          a=a-b;
                    else
                          b=b-a;
              }
              t3.setText("GCD :"+a);
        }
        if(ae.getSource()==b2)
        {
              t1.setText("");
              t2.setText("");
               t3.setText("");
              t1.requestFocus();
        }
    }

public static void main(String[] args){
JFrame frame = new JFrame(" JFrame");
GcdS gcd = new GcdS();
frame.add(gcd);
```

```
frame.setSize(800,400);
frame.setDefaultCloseOperation(frame.EXIT_ON_CLOSE);
gcd.init();
frame.setVisible(true);
 }
}
```

3.GCD-Frame

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;

/*<applet code=Gcddemo width=200 height=200></applet>*/

public class Gcddemo extends Frame implements ActionListener{
     Label l1,l2,l3;
     TextField t1,t2,t3;
     Button b1,b2;

     Gcddemo()
     {
      l1=new Label("Num 1");
      l2=new Label("Num 2");
      l3=new Label("answer");
      t1=new TextField();
      t2=new TextField();
      t3=new TextField();
      b1= new Button("Calculate");
      b2=new Button("Clear");
      b1.addActionListener(this);
      b2.addActionListener(this);

      setLayout(new GridLayout(4,2));
      setSize(400,200);
      setVisible(true);

      add(l1);
      add(t1);
      add(l2);
      add(t2);
      add(l3);
      add(t3);
```

```
        add(b1);
        add(b2);

        this.addWindowListener( new WindowAdapter()
        {
              public void windowClosing(WindowEvent We)
              {
                    System.exit(0);
              }
       });

 }

 public void actionPerformed(ActionEvent ae)
 {
        if(ae.getSource()==b1)
        {
              int a=Integer.parseInt(t1.getText());
              int b=Integer.parseInt(t2.getText());

              while(a!=b)
              {
                    if(a>b)
                          a=a-b;
                    else
                          b=b-a;
              }
              t3.setText("GCD :"+a);
       }

        if(ae.getSource()==b2)
        {
              t1.setText("");
              t2.setText("");
               t3.setText("");
              t1.requestFocus();
        }
 }

 public static void main(String[] args)
 {
        new Gcddemo();
}
}
```

4.Fibnoacci -Applet

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;

/*<applet code=FibA width=200 height=200></applet>*/

public class FibA extends Applet implements ActionListener{
      Label l1,l2;
      TextField t1,t2;
      Button b1,b2;

 public void init(){
       l1=new Label("Num 1");
       l2=new Label("answer");
       t1=new TextField();
       t2=new TextField();
       b1= new Button("Calculate");
       b2=new Button("Clear");
       b1.addActionListener(this);
       b2.addActionListener(this);

       setLayout(new GridLayout(4,2));

       add(l1);
       add(t1);
       add(l2);
       add(t2);
```

```
      add(b1);
      add(b2);
}

public void actionPerformed(ActionEvent ae)
{
      if(ae.getSource()==b1)
      {
            int a=fib(Integer.parseInt(t1.getText()));
            t2.setText(String.valueOf(a));
    }
}

      public int fib(int n)
      {
            if(n==0 || n==1)
                  return n;
            else
                  return fib(n-1) + fib(n-2);
      }
}
```

---

5. Factorial -Swing

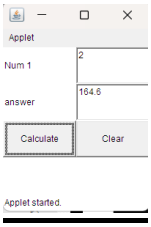
```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

/*<applet code=Facts width=200 height=200></applet>*/

public class Facts extends JApplet implements ActionListener{
 JLabel l1,l2;
 JTextField t1,t2;
 JButton b1,b2;
```

```
 public void init()
{
  l1=new JLabel("Num 1");
  l2=new JLabel("answer");
  t1=new JTextField();
  t2=new JTextField();
  b1= new JButton("Calculate");
  b2=new JButton("Clear");
  b1.addActionListener(this);
  b2.addActionListener(this);

  Container cp=getContentPane();
  cp.setLayout(new GridLayout(4,2));

  add(l1);
  add(t1);
  add(l2);
  add(t2);
  add(b1);
  add(b2);

 }

 public void actionPerformed(ActionEvent ae)
 {
  if(ae.getSource()==b1)
  {
        int a=fact(Integer.parseInt(t1.getText()));
        t2.setText(String.valueOf(a));
     }
 }

  public int fact(int n)
  {
        if(n==0)
              return 1;
        else
              return fact(n-1)*n;
  }


 public static void main(String[] args)
{
JFrame frame = new JFrame(" JFrame");
Facts fac = new Facts();
frame.add(fac);
frame.setSize(800,400);
frame.setDefaultCloseOperation(frame.EXIT_ON_CLOSE);
```

```
      fac.init();
      frame.setVisible(true);

       }
      }
```

---

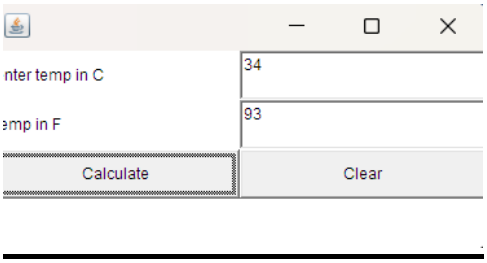
6. US -Doller – Applet




```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;

/*<applet code=inr width=200 height=200></applet>*/

public class inr extends Applet implements ActionListener{
      Label l1,l2;
      TextField t1,t2;
      Button b1,b2;


 public void init(){
       l1=new Label("INR ");
       l2=new Label("US DOLLER");
       t1=new TextField();
       t2=new TextField();
       b1= new Button("Calculate");
```

```
        b2=new Button("Clear");
        b1.addActionListener(this);
        b2.addActionListener(this);

        setLayout(new GridLayout(4,2));

        add(l1);
        add(t1);
        add(l2);
        add(t2);
        add(b1);
        add(b2);

 }

 public void actionPerformed(ActionEvent ae)
 {
        if(ae.getSource()==b1)
        {
              float f1,f2;
              f1=Float.parseFloat(t1.getText());
              f2=f1*82.30f;
              t2.setText(Float.toString(f2));
        }
              if(ae.getSource()==b2)
        {
              t1.setText("");
              t2.setText("");
              t1.requestFocus();
        }
 }
}
```

7. Celcius -Frame

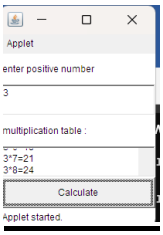



```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;

/*<applet code=cel width=200 height=200></applet>*/

public class cel extends Frame implements ActionListener{
 Label l1,l2;
 TextField t1,t2;
 Button b1,b2;


  cel()
{
  l1=new Label("enter temp in C");
  l2=new Label("temp in F");
  t1=new TextField();
  t2=new TextField();
  b1= new Button("Calculate");
  b2=new Button("Clear");
  b1.addActionListener(this);
  b2.addActionListener(this);

  setSize(400,200);
  setLayout(new GridLayout(4,2));
  setVisible(true);

  add(l1);
  add(t1);
  add(l2);
  add(t2);
  add(b1);
  add(b2);
```

```
        addWindowListener( new WindowAdapter(){
              public void WindowClosing(WindowEvent We){
                    System.exit(0);
              }
            });

       }

       public void actionPerformed(ActionEvent ae)
       {
        if(ae.getSource()==b1)
        {
              int C=Integer.parseInt(t1.getText());

              int F=(C*9/5)+32;
              t2.setText(String.valueOf(F));
            }
        if(ae.getSource()==b2)
        {
              t1.setText("");
              t2.setText("");
              t1.requestFocus();
        }

      }


        public static void main(String[] args)
        {
              new cel();
        }
       }
```

---

8. Multiplication Table – Applet

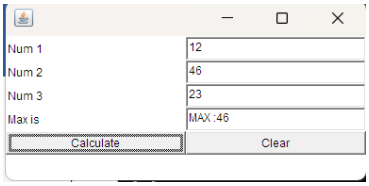



```
import java.awt.*;

import java.awt.event.*;

import java.applet.*;
```

```
/*<applet code=multi width=200 height=200></applet>*/

public class multi extends Applet implements ActionListener{
      Label l1,l2;
      TextField t1;
      TextArea txt1;
      Button b1;


 public void init(){
       l1=new Label("enter positive number");
       l2=new Label("multiplication table :");
       t1=new TextField();
       txt1=new TextArea(10,20);
       b1= new Button("Calculate");
       b1.addActionListener(this);


       setLayout(new GridLayout(5,4));


       add(l1);
       add(t1);
       add(l2);
       add(txt1);
       add(b1);

 }

 public void actionPerformed(ActionEvent ae)
 {
       if(ae.getSource()==b1)
```

```
        {

              int n=Integer.parseInt(t1.getText());
              int res;
              txt1.setText("");

              for(int i=1;i<=10;i++)
              {
                    res=n*i;
              txt1.append(n+ "*" + i + "=" +res+ "\n");
              }

        }
  }
}
```

9. Largest of Three number -Frame




```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;

/*<applet code=largest width=200 height=200></applet>*/

public class largest extends Frame implements ActionListener{
 Label l1,l2,l3,l4;
 TextField t1,t2,t3,t4;
```

```
Button b1,b2;


largest(){
 l1=new Label("Num 1");
 l2=new Label("Num 2");
  l3=new Label("Num 3");
  l4=new Label("Max is ");
 t1=new TextField();
 t2=new TextField();
 t3=new TextField();
 t4=new TextField();
 b1= new Button("Calculate");
 b2=new Button("Clear");
 b1.addActionListener(this);
 b2.addActionListener(this);

 setSize(400,200);
 setLayout(new GridLayout(6,6));
 setVisible(true);

 add(l1);
 add(t1);
 add(l2);
 add(t2);
 add(l3);
 add(t3);
 add(l4);
 add(t4);
 add(b1);
 add(b2);

 this.addWindowListener( new WindowAdapter()
 {
       public void windowClosing(WindowEvent We)
       {
             System.exit(0);
       }
    });

}

public void actionPerformed(ActionEvent ae)
{
 if(ae.getSource()==b1)
 {
       int n1,n2,n3,max;
       n1=Integer.parseInt(t1.getText());
       n2=Integer.parseInt(t2.getText());
       n3=Integer.parseInt(t3.getText());
```

```
            if(n1>n2 && n1>n3)
                  max=n1;
            else if(n2>n1 && n2>n3)
                  max=n2;
            else
                  max=n3;
        t4.setText("MAX :"+max);
      }
       if(ae.getSource()==b2)
      {
            t1.setText("");
            t2.setText("");
             t3.setText("");
             t4.setText("");
            t1.requestFocus();
      }
     }

      public static void main(String[] args)
      {
            new largest();
      }
     }
```

10. Protected and private package import

```
package mp;
public class parent {
    protected int a = 20;
    private int b = 10;

    protected void protectmeth() {
        System.out.println("this is proected method");

    }

    private void privatemeth() {
        //System.out.println("this is private method");
```

```
    }
}
```

Javac -d . parent.java

```
package mp1;
import mp.parent;

public class child extends parent {
    public void displaymod() {
        System.out.println("Protected value :" + a);
        //System.out.println("private value " + b);

        protectmeth();
        //privatemeth();
    }
}
```

Javac -d .child.java

```
import mp1.child;
public class main {
    public static void main(String[] args) {
        child c = new child();
        c.displaymod();
    }
}

Javac main.java
Java main
```