D. S. COLLEGE

ALIGARH

Java Assement of Applet programming

Session- 2023-2024

Submitted By –
Mudit Sharma
Arshpreet

Submitted to -
Dr.Monika Varshney

Applet programming

Applets are small java program that re primarily used in internet Computing. They can be transported one computer yo another and run using the applet viewer and web browser that support java .

It can perform arithmetic operations , display graphics, play sound , accept input user , create animation and so on.

Applets are two types:-

Local applet.

Remote applet.

Local applet:- An applet developed locally and stored in a local system is known as a local applet.

Local applet is only run on own computer without internet connection.

Remote applet:- A remote applet is that which is developed by someone else and stored on a remote .

For an access or download remote applet you need an internet connection.

Applet vs Application.

Applet doesn't have main () method while application have main () method.

Applet cannot run independently. It needs web page using a specific feature known as HTML tag.

Applet cannot read from or write to the file in local system.

Applets cannot communicate with other servers on the network.

Applets cannot run a program from local computer .

Applets are restricted from using other libraries from other language like C/C++ etc.[But not that java supports the same thing using native methods.]

why we need Applets?

When we need to include something dynamic to include in our web page .

When we require some flash outputs like applet to produce some sounds , animation ,and some special effects when displaying some certain pages .

When we want to create a program and want it to make available on internet so that it could be used by others .

Life Cycle

Every java applet has a life cycle through which it passes throughout his life span. Every java applet inherits the behaviour from the Applet class. Applet like conatins the following states.

Born or initialization.

Running state

Idle state

Dead or destroyed state.

1) Born Or Initialization state :- when an applet is loaded it enters the initialization state . this happen by calling the init() method .initialization occurs only once in the applet's life cycle . To go with any of the above points we need to override init() method.

Syntax:-]

Public void init()

{________

__________(Action)

}

Create object as required by the applet.

Set up initial values

Load images or fonts .

Set up colors .

2. Running State:- Applet enters running state when the system calls the start() method of applet class . this method oocurs more than once .

Syntax :-

```
public void start()

{________

,_________(Action)

}
```

Idle or Stopped state :- an applet becomes idle when it is stopped from running .

```
public void Stop ()

{________

___________(Action)

}
```

Dead State :- An applet is said yo be dead when it is removed from the memory . this occur automatically by involving the destroy() method . like initialization and destroy occurs only once in the applet life cycle .

Syntax:-

```
Public void destroy ()

{

________

___________(Action)

}
```

Display :- whenever an applet has to perform some output operation . it enters display state . this happens after an applet enters to the running state . the paint() method is called to accomplish the task .

Syntax:-

```
public void paint(Graphics g)

{
```

____

_____________(Action)

}

Note:- display state is not considered as the part of life cycle of an applet.

Writing java applet code

To create an applet we follows:-

Building an applet code (.java file)

Creating an executable applet (.class file)

Designing a webpage using HTML tags .

Preparing <Applet> tag.

Incorporating <Applet> tag into the webpage.

Creating HTML file .

Testing the applet code .

Running the applet code .

In order to run the applet . we must all these file under the same directory.

File name .java

Filename.class

Filename.html

Now in order to run applet. We have two ways :-

Using java enable web browser.

Using java applet viewer.

Running the applet (using java enable web browser**)**

If we use a java enabled web browser, we will be able to see the entire web page containing the applet. If we use the applet viewer tool . we only see the applet output.

The applet viewer is available as a part of java development kit that we have been using so far .

we can use it to run our applet as follows -

Applet viewer filename. HTML

Here you need to note that the file name is with extension .html &not .java not .class aa applet viewer runs the applet which is present in the form of .HTML file.

Error :-

When we write a program , we may make some common mistake while typing the program .

The error are broadly categorized into :-

Compile Time Error.

Run Time Error.

Compile Time error:- All syntax error that can be detected and displayed by the java compiler are called compile time error.

Run time error:- Sometime it happens that our program compile successfully but when we run it gives us storage result that is unexpected result . this may be due to the wrong logic of the program and hence our program produces unexpected results. The most common runtime error are :-

Divide by zero.

Accessing an element beyond index array.

The file that we need , don't exist.

Storing wrong type of values in array and etc.

Exception:-

An exception is a condition that is caused by the run-time error. When java interpreter encounters runtime error viz divide by zero . it creates an exception object and thorws it.

Now , if we have written the code to catch the exceptions and handle it properly, then our program will keep running , otherwise the whole program will collapse .

So, the process of catching the exception thrown by the error condition and display an appropriate message for the some is called **"Exception handling".**

The purpose of exception handling mechanism is to provide a means to detect and report and exceptional circumstances . so that appropriate action can be taken . the mechanism suggests incorporation separate eroror handling code that perform the following task :-

Find the exceptions (Hit the exceptions)

Inform the error (throw the exception)

Receive the error (catch the exception)

Take action(Handle the exception)

Exveption in java can be handled into two ways :-

Checked Exception :- these exception are handled in the code itself with the help of try. Catch blocks Checked Exception are extended from java.lang.exception.

Unchecked Exception:- these exception are not essentially handled in the program code . The JVM handle such exceptions . Unchecked exception are extended from the java.lang.runtime exception class .

M.C.Q

Q1. Java programming that are primarily used for internet commuting.

a) java b) XML c) SQL d) Applet

Ans:- d)Applet.

Q2.Applets can perform :-

a) Arithmetic operations b) Display Graphics

c) Play Sound. d) Create animation

Answer :- all the above

Q3. In order to load remote applet we must know

a) Internet protocol address b) Internet

c) Applet address d) none of the above

Ans:- c) Applets address (U.R.L)

Q4. Which doesn't need main() function in java ?

a) java b) Application c) Applet d) SQL

Ans:- c) Applet.

Q5._________can make something dynamic to includes in our web page .

a) Applet b) java c) HTML d) c++

Ans:- Applet.

Q6._____________tag to initialise the applet into HTML code.

a) <style> tag b)<applet java> tag c) <applet > tag d) All of these.

Ans:- c) <Applet> tag.

Q7.Applet viewer runs the applet file in form of ______

a).doc file b).html file c) .XML file d) .gif file

Ans:- b) .html file.

Q8. Method to Initialization applet class in java ?

a) start() b) init() c) drawstring() d) display()

Ans:- b) init()

Q9.How many part of Applet life cycle?

a) 7 b) 5 c) 4 d) 6

Ans:- c) 4

Q10. Parts of life cycle of an Applet are

a) Born or initialization state b) Running state

c) Idle state d) Dead and Destroyed state

Ans :- e) All of these

Q11.How to display the output of an Applet?

a) Applet viewer b) web browser c) both a & b d) none of these

Ans:- c) Applet viewer and web browser.

Q12. Applet is based on ______

a) system based b) java based c) client sever based d) Web based

Ans:- d) Web based.

Q13.Types of Applet are _____

a) both d & c b) private applet c) remote applet d) local applet

Ans:- local applet and remote applet.

Q14.An applet developed locally and stored in a local system is known as __________

a) Global b) Local c) Private d) Public

Ans:- b) Local applet.

Q15.Which method apply only once in applet's life cycle .

a) Initialization state b) dead state c) only a d) both a & b.

Ans:- d) initialization state and dead state.

Q16. ______Method to destroy the applet class in java .

a) start() b) destroy() c) stop() d) transient()

Ans:- b) destroy()

Q17._______method to stopped the applet class in java .

a) stop() b)paint() c) drawstring() d) transient()

Ans:- a) stop()

Q18._______method of running state in applet class in java .

a) init() b)paint() c) display() d) Start()

Ans:- d) Start()

Q19.________method is called to accomplish the task.

a) init() b)paint() c) action() d) task()

Ans:- paint()

Q20.Exception in java can be categorized into:-

a) Checked b) Unchecked c) Both a & b d) null

Ans:- c) Both Checked and unchecked Exception.

Q21.Whenever an applet has to perform some output operation called _____.

a) Output b) Print c) Display d) Result

Ans.:- c) Display

Q22. Which state is not the part of life cycle of an Applet.

a) Start b) Destroy c) Display d) Stop

Ans:- c) Display.

Q23.________condition that is caused by the runtime error.

a) Exception b) Error c) only a d) only b

Ans:- c) Exception

Q24. When we write a program , we may make some common mistake while typing the program is called _______.

a) exception b) error c) compile time error d) run time error

Ans:- Error

Q25. Error can be categorized into.___&___.

a) exception b) Compile time error c) run time error d) both b & c.

Ans:- d) Compile error and Run-time error.

Q26. Which one of the following is a void declaration of an Applet?

a) public class Myclass extends Applet { }

b) public class Myclass.java.html.applet extends java.applet.Applet{ }

c) none of the above

d) all of the above

Answer :- a) public class Myclass extends Applet { }

Q27. Which of these function is called to display the output ?

a) display() b)paint() c) displayapplet(). d)printapplet()

Answer :- b) paint()

Q28. Which of these methods can be used to output a string in an Applet?

a) display() b)paint() c) drawstring() d) transient()

Answer :- c) drawstring()

Q29. Which of the following method is a part of java.applet.awt?

a) display() b) paint() c) displayapplet(). d) transient()

Answer:- b) paint()

Q30. Awt stand for _____________.

a) abstract windowing toolkit

b) abstract widow transient.

c) applet widow terminal

d) none of these .

Answer:- a) abstract windowing toolkit.

Q31. Which of these modifiers can be used for a variable so that it can be accessed from any thread or parts of a program.

a) transient b)volatile c) global d)none of these

Answer :- b) volatile

Q32. Which of these operators can be used to get run information about the object ?

a) getinfo b) info c) instanceof d)getinoof

Answer:- c) instanceof

Q33. Which are the common security restriction in applets ?

a) Applet can't load libraries or define native methods .

b) An Applet can't read every system properties.

c) Applet can play sounds .

d) Both a & b .

Answer:- d) both a & b.

Q34. From the following statements which is a drawback for applet ?

a) It works at client side so less response time.

b) It can be executed by browser running under many platforms, including linus, window & Mac OS etc.

c) Secured .

d) Plugin is required at client browser to execute applet.

Answer :- d) Plugin is required at client browser to execute applet.

Q35. What is used to run an Applet?

a) Html file b) Apllet viewer tool c) Both a & b d) none of these

Answer :- a) HTML file .

Q36.when an applet begins , in which sequence will the awt call the method ?

a) init() b) start() c) paint() d) all of these.

Answer :- d) all of these

Q37. Which method is first called for any applet when it starts its execution.

a) void init() b) void destroy() c) Boolean is Active(). d) none of these

Answer:- a) void init()

Q38. Java Applet define how many interfaces ?

a) 8 b) 3 c) 4 d) 6

Answer :- b) 3

Q39. Which is invoked after the init() method or browser is maximized ?

a) public void start(). b) public void paint(Graphics g)

c) public void stop() d) public void init()

Answer :- public void start()

Q40. Which applet java.awt.component class provide the life cycle method ?

a) public void paint(Graphics g) b) public void destroy()

c) public void stop() d) public void init()

Answer:- a) public void paint(Graphic g)

Q41. Applet can be embedded in a ________.

a) MS word b) RTF file c) Gif file d) HTML file

Answer :- d) html file

Q42. The class at top of the exception class hierarchy is ________.

a) object b) exception c) arithmetic exception d) throwable

Answer :- d) throwable

Q43. In which of the following package exception class exist ?

a) java.util b) java.file c) java.net d) java.lang

Answer :-c) java.net

Q44. Exception generated in try block is caught in __________block.

a) Catch b) Throw c) Throws d) Finally

Answer:- c) Throws

Q45. Which keyword is used to explicitly throw an exception ?

a) try b) catch c) throwing d) throw

Answer :- d) throw

Q46. Which exception is thrown when divide by zero statement ?

a) ArithmeticException b) NumberFormatException

c) NullPointerException . d) None of these

Answer:- a) ArithmeticException

Q47. Which of the following blocks execute compulsory whether exception is caught or not .

a) finally b) catch c) throws d) throw

Answer:- a) finally

Q48. Exception is a _____ ?

a) exception b) error c) compile time error d) run time error

Answer :- d) run time error

Q49. Which of these is not a part of exception handling?

a) finally b) catch c) thrown d) try

Answer:- c) thrown

Q50. Which of these keywords must be used to monitor for exceptions?

a) finally b) thrown c) catch d) try

Answer :- d) try

Q51. Which of these class is related to all the exceptions that cannot be caught ?

a) error b) exception c) run time error d) all of these

Answer :- a) error

Q52. Which of these operator is used of an exception thrown by using throw ?

a) new b) malloc c) alloc d) thrown

Answer :- a) new

Q53.Which of these keywords is uses to by the calling function to guard against the except that is thrown by called function ?

a) throws b) throw c) catch d)finally

Answer:- throws

Q54. A single try block must be followed by which of these ?

a) finally b) catch c) none d) both a & b.

Answer :- d) both a & b.

Q55.Which of these keywords must be used to handle the exception thrown by try block in rational manner ?

a) finally b) throw c) catch d) try

Answer :-c) catch

Q56. Package of exception class exist ?

a) java.lang b) java.util c) java.I.O d) java.file

Answer :- a) java.lang

Q57. Which of these is a Super Class of all errors & exceptions in the java languages?

a) catchfile b) compile time exception c) run time exception d) none

Answer c) runtime exception

Q58.which of these handle the exception when no catch is uses ?

a) Default handler b) finally thrown

c) throw handler d) java runtime system

Answer :- d) java runtime system

Q59. What exception thrown by parse Int() method?

a) ArithmeticException. b) ClassNotFoundException

c) NullPointerException. d) NumberFormatException

Answer :- d) NumberFormatException

Q60. What is length of the application box made by following java program -

```
import java.awt.*;
import java.applet.*;
public class Myapplet extends Applet
{
    public void paint( Graphics g )
    {
       g.drawstring(" A first applet program ", 20 20 );
     }
}
```

a) 20 , 20

b) 20

c) Compile time error

d) Run time error

Answer :- 20

Q61. What is the message is displayed in the applet made by following java program -

```
import java.awt.*;
import java.applet.*;
public class Myapplet extends Applet
```

```
    {
        public void paint( Graphics g )
        {
            g.drawstring(" A first applet program ", 20 20 );
         }
    }
```

a) A first applet program 20 20 .

b) A first applet program

c) Compile time error

d) Run time error

Answer :- b) A first applet program

Q. 62 What will be output for the following code?

```
class Test extends Exception { }
  class Main {
   public static void main(String args[]) {
      try {
         throw new Test();
      }
      catch(Test t) {
         System.out.println("Got the Test Exception");
      }
      finally {
         System.out.println("Inside finally block ");
      }
  }
}
```

A. Got the Test Exception Inside finally block

B. Got the Test Exception

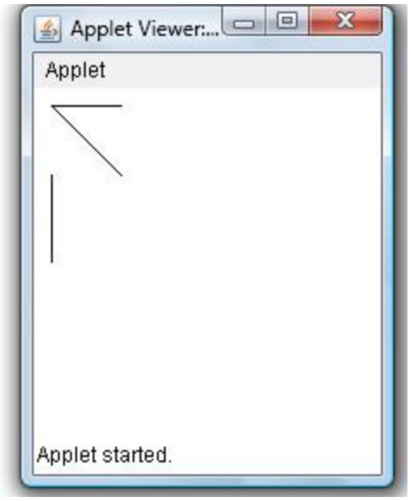
C. Compiler Error

D. Inside finally block

Ans : a) Got the Test Exception Inside finally block

Program to draw a line in applet in a java .

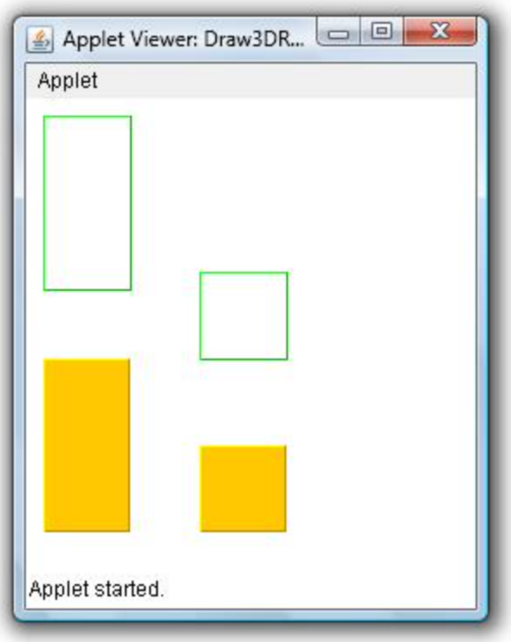
```
import java.applet.Applet;

import java.awt.Graphics;

public class DrawLineExample extends Applet

{

      public void paint(Graphics g)

        {

        g.drawLine(10,10,50,50);

       //draw vertical line

       g.drawLine(10,50,10,100);

      //draw horizontal line

       g.drawLine(10,10,50,10);

       }

}
```



Applet started.

Write a program to draw a rectangle in Applet java programming.

```
import java.applet.Applet;

import java.awt.Color;

import java.awt.Graphics;

public class Draw3DRectanglesExample extends Applet

{

    public void paint(Graphics g)

    {

     g.setColor(Color.green);

    //this will draw a 3-D rectangle of width 50 & height 100 at (10,10)

     g.draw3DRect(10,10,50,100,true);

    //this will draw a 3-D square

    g.draw3DRect(100,100,50,50,true);

    g.setColor(Color.orange);

    //this will draw a filled 3-D rectangle of width 50 & height 100 at (10,10)

    g.fill3DRect(10,150,50,100,true);

    //this will draw a filled 3-D square

    g.fill3DRect(100,200,50,50,true);

      }

}
```



Applet started.

Write a program to print the sum of two number in applet java programming.

Step 1: Create the java program with " filename.java ".

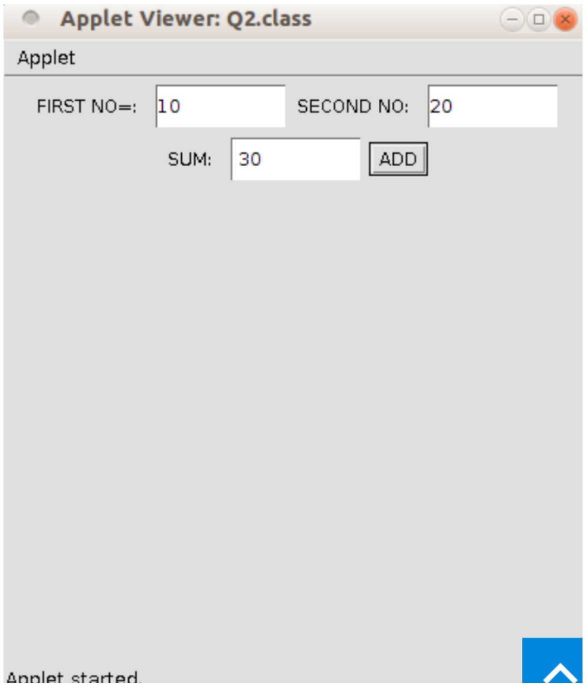
Step 2: Create the html program with "filename.html".

Step 3: Compile the java program "javac filename.java".

Step 4: View applet using "appletviewer filename.html".

```
import java.awt.*;

import java.awt.event.*;

import java.applet.*;

public class Q2 extends Applet implements ActionListener

{

   TextField t1 = new TextField(10);

   TextField t2 = new TextField(10);

   TextField t3 = new TextField(10);

   Label l1 = new Label("FIRST NO=:");

   Label l2 = new Label("SECOND NO:");

   Label l3 = new Label("SUM:");

   Button b = new Button("ADD");

   public void init()

   {

      t1.setForeground(Color = Red);

      add(l1);

      add(t1);
```

```
        add(l2);
        add(t2);
        add(l3);
        add(t3);
        add(b);
        b.addActionListener(this);
    }
public void actionPerformed(ActionEvent e)
    {
        if (e.getSource() == b)
        {
            int n1 = Integer.parseInt(t1.getText());
            int n2 = Integer.parseInt(t2.getText());
            t3.setText(" " + (n1 + n2));
        }
    }
}
<HTML>
     <HEAD>
        <TITLE>WELCOME TO  JAVA APPLET</TITLE>
    </HEAD>
 <BODY>
        <CENTER>
            <H1>WELCOME TO THE APPLET</H1> </CENTER>
        <BR>
        <APPLET CODE=Q2.class WIDTH=400 HEIGHT=400> </APPLET>
    </BODY>
</HTML>
```



FIRST NO=: 10 SECOND NO: 20

SUM: 30

Applet started.

Write a program to check whether input number is odd or even.

Step 1: Create the java program with " filename.java ".

Step 2: Create the html program with "filename.html".

Step 3: Compile the java program "javac filename.java".

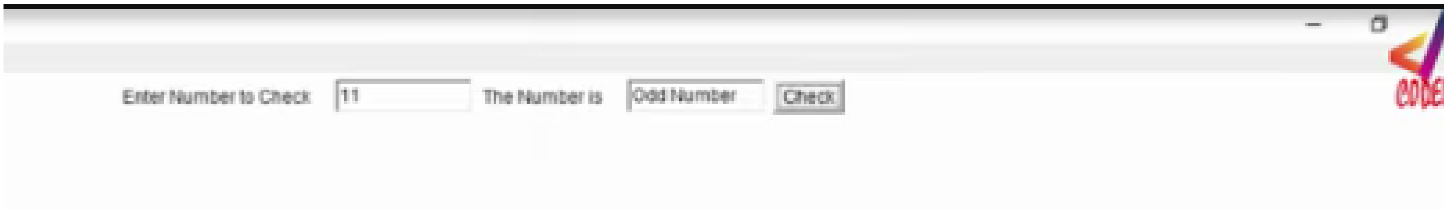
Step 4: View applet using "appletviewer filename.html".

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
public class oddeven extends Applet implements ActionListener
{
   TextField t1 = new TextField(10);
   TextField t2 = new TextField(10);
   Label l1 = new Label("FIRST NO=:");
   Label l2 = new Label("SECOND NO:");
   Button b = new Button("check");
   public void init()
   {
      t1.setForeground(Color = Red);
      add(l1);
      add(t1);
      add(l2);
```

```
    add(t2);

    add(b);

    b.addActionListener(this);

  }

public void actionPerformed(ActionEvent e)

  {

    if (e.getSource() == b)

    {

      int n1 = Integer.parseInt(t1.getText());

If(n%2==0)

      t2.getText(" Even number ");

else

      t2.getText(" Even number ");


    }

  }

}
```

```
<HTML>

   <HEAD>

    <TITLE>WELCOME TO  JAVA APPLET</TITLE>

  </HEAD>

 <BODY>
```

```
    <CENTER>
      <H1>WELCOME TO THE APPLET</H1> </CENTER>
    <BR>
    <APPLET CODE=oddeven.class WIDTH=400 HEIGHT=400> </APPLET>
  </BODY>
</HTML>
```



Enter Number to Check 11 The Number is Odd Number

Write a program to check whether input number is prime or not.

Step 1: Create the java program with " filename.java ".

Step 2: Create the html program with "filename.html".

Step 3: Compile the java program "javac filename.java".

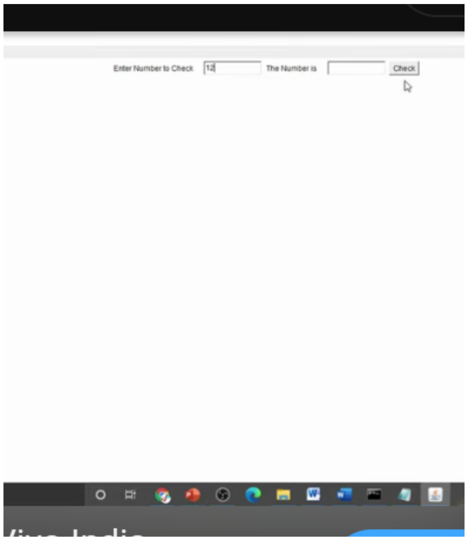
Step 4: View applet using "appletviewer filename.html".

```
import java.awt.*;

import java.awt.event.*;

import java.applet.*;

public class oddeven extends Applet implements ActionListener

{

   TextField t1 = new TextField(10);

   TextField t2 = new TextField(10);

   Label l1 = new Label("FIRST NO=:");

   Label l2 = new Label("SECOND NO:");

   Button b = new Button("check");

   public void init()

   {

      t1.setForeground(Color = Red);

      add(l1);
```

```
        add(t1);

        add(l2);

        add(t2);

        add(b);

        b.addActionListener(this);

    }

public void actionPerformed(ActionEvent e)

    {

        if (e.getSource() == b)

        {

            int n1 = Integer.parseInt(t1.getText());

            int count =0;

for(int i=1; i<=n1; i++)

{

If(n%i==0)

Count+=1;

}

if(count==2)

            t2.getText(" Prime number. ");

else

            t2.getText(" Composite Number  ");


        }

    }
```

```
}
```

```
<HTML>
     <HEAD>
        <TITLE>WELCOME TO  JAVA APPLET</TITLE>
    </HEAD>
 <BODY>
        <CENTER>
            <H1>WELCOME TO THE APPLET</H1> </CENTER>
        <BR>
        <APPLET CODE=oddeven.class WIDTH=400 HEIGHT=400> </APPLET>
    </BODY>
</HTML>
```



Enter Number to Check 12 The Number is

BorderLayout (LayoutManagers)

Java LayoutManagers

The LayoutManagers are used to arrange components in a particular manner. The Java LayoutManagers facilitates us to control the positioning and size of the components in GUI forms. LayoutManager is an interface that is implemented by all the classes of layout managers. There are the following classes that represent the layout managers:

java.awt.BorderLayout
java.awt.FlowLayout
java.awt.GridLayout
java.awt.CardLayout
java.awt.GridBagLayout
javax.swing.BoxLayout
javax.swing.GroupLayout
javax.swing.ScrollPaneLayout
javax.swing.SpringLayout etc.
Java BorderLayout

The BorderLayout is used to arrange the components in five regions: north, south, east, west, and center. Each region (area) may contain one component only. It is the default layout of a frame or window. The BorderLayout provides five constants for each region:

public static final int NORTH

public static final int SOUTH

public static final int EAST

public static final int WEST

public static final int CENTER

Constructors of BorderLayout class:

BorderLayout(): creates a border layout but with no gaps between the components.

BorderLayout(int hgap, int vgap): creates a border layout with the given horizontal and vertical gaps between the components.

Example of BorderLayout class: Using BorderLayout() constructor

FileName: Border.java

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

public class Border
{
 JFrame f;
Border()
{
    f = new  JFrame();

     // creating buttons
     JButton b1 = new  JButton("NORTH");; // the button will be labeled as NORTH
     JButton b2 = new  JButton("SOUTH");; // the button will be labeled as SOUTH
     JButton b3 = new  JButton("EAST");; // the button will be labeled as EAST
     JButton b4 = new  JButton("WEST");; // the button will be labeled as WEST
     JButton b5 = new  JButton("CENTER");; // the button will be labeled as CENTER

    f.add(b1, BorderLayout.NORTH); // b1 will be placed in the North Direction
    f.add(b2, BorderLayout.SOUTH);  // b2 will be placed in the South Direction
    f.add(b3, BorderLayout.EAST);  // b2 will be placed in the East Direction

    f.add(b4, BorderLayout.WEST);  // b2 will be placed in the West Direction

    f.add(b5, BorderLayout.CENTER);  // b2 will be placed in the Center


    f.setSize(300, 300);

    f.setVisible(true);

}

public static void main(String[] args) {

    new Border();

}

}
```

Output:

download this example

Example of BorderLayout class: Using BorderLayout(int hgap, int vgap) constructor

The following example inserts horizontal and vertical gaps between buttons using the parameterized constructor BorderLayout(int hgap, int gap)

FileName: BorderLayoutExample.java

```
// import statement

import java.awt.*;

import javax.swing.*;
```

```
public class BorderLayoutExample
{
 JFrame jframe;
// constructor
BorderLayoutExample()
{
    // creating a Frame
    jframe = new JFrame();
    // create buttons
     JButton btn1 = new JButton("NORTH");
     JButton btn2 = new JButton("SOUTH");
     JButton btn3 = new JButton("EAST");
     JButton btn4 = new JButton("WEST");
     JButton btn5 = new JButton("CENTER");
     // creating an object of the BorderLayout class using
     // the parameterized constructor where the horizontal gap is 20
     // and vertical gap is 15. The gap will be evident when buttons are placed
     // in the frame
    jframe.setLayout(new BorderLayout(20, 15));
    jframe.add(btn1, BorderLayout.NORTH);
    jframe.add(btn2, BorderLayout.SOUTH);
    jframe.add(btn3, BorderLayout.EAST);
    jframe.add(btn4, BorderLayout.WEST);
    jframe.add(btn5, BorderLayout.CENTER);
    jframe.setSize(300,300);

    jframe.setVisible(true);

}

// main method

public static void main(String argvs[])

{

    new BorderLayoutExample();

}

}
```

Output:

Java BorderLayout: Without Specifying Region

The add() method of the JFrame class can work even when we do not specify the region. In such a case, only the latest component added is shown in the frame, and all the components added previously get discarded. The latest component covers the whole area. The following example shows the same.

FileName: BorderLayoutWithoutRegionExample.java

```
// import statements

import java.awt.*;

import javax.swing.*;


public class BorderLayoutWithoutRegionExample
```

```
{
JFrame jframe;

// constructor
BorderLayoutWithoutRegionExample()
{
   jframe = new JFrame();

   JButton btn1 = new JButton("NORTH");
   JButton btn2 = new JButton("SOUTH");
   JButton btn3 = new JButton("EAST");
   JButton btn4 = new JButton("WEST");
   JButton btn5 = new JButton("CENTER");

   // horizontal gap is 7, and the vertical gap is 7
   // Since region is not specified, the gaps are of no use
   jframe.setLayout(new BorderLayout(7, 7));

   // each button covers the whole area
   // however, the btn5 is the latest button
   // that is added to the frame; therefore, btn5
   // is shown
   jframe.add(btn1);
   jframe.add(btn2);
```

```
    jframe.add(btn3);
    jframe.add(btn4);
    jframe.add(btn5);

    jframe.setSize(300,300);
    jframe.setVisible(true);
}

// main method
public static void main(String argvs[])
{
    new BorderLayoutWithoutRegionExample();
}
}
```

Output: