**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| S. No. | Name of Program | Date of Implementation | Remarks |
| 1. | Find total, average of two numbers | 13/01/2022 |  |
| 2. | Classes and objects | 24/01/2022 |  |
| 3. | constructor | 27/01/2022 |  |
| 4. | Type conversion | 07/02/2022 |  |
| 5. | (+,-,\*,/) operations on complex numbers. | 13/02/2022 |  |
| 6. | Inheritance | 20/02/2022 |  |
| 7. | Polymorphism | 26/02/2022 |  |
| 8. | Method overriding | 3/03/2022 |  |
| 9. | Key adapter and mouse adapter | 11/03/2022 |  |
| 10. | Array manipulation | 31/03/2022 |  |
| 11. | Layout managers | 7/04/2022 |  |

**ASSIGNMENT: 1**

# Program to find total, average of given two numbers by using function with default arguments, static data members and this keyword

CODE:

import java.util.\*;

class javalab1{

int x,y;

javalab1(int a,int b){

this.x=a;

this.y=b;

System.out.println("sum= "+(x+y));

}

static double average(int num1,int num2){

return (num1+num2)/2.0;

}

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

System.out.print("Enter first number- ");

int a=sc.nextInt();

System.out.print("Enter second number- ");

int b=sc.nextInt();

javalab1 obj=new javalab1(a,b);

System.out.println("average= " +average(a,b));

}

}

**Output🡪**

Enter first number- 10

Enter second number- 20

sum= 30

average= 15.0

**ASSIGNMENT 2**

# Program to illustrate class and objects.

CODE:

//illustrate class and object

import java.util.\*;

//class

class labs{

int lab\_no;

String work;

}

public class assign2 {

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

//object

labs obj=new labs();

System.out.println("enter lab number and work assigned:" );

obj.lab\_no=sc.nextInt();

Scanner in =new Scanner(System.in);

obj.work=in.nextLine();

System.out.println("Lab number= "+obj.lab\_no);

System.out.println("Work= "+obj.work);

}

}

Output🡪

enter lab number and work assigned:

2

implementation of objects and classes in java.

Lab number= 2

Work= implementation of objects and classes in java.

**ASSIGNMENT 3**

# Program to illustrate constructors.

CODE:

//illustrate constructor

class assign3 {

int roll;

String name;

//constructor overloading

assign3(){//default

}

assign3(int r,String n){//parameterized

roll=r;

name=n;

System.out.println(r+" "+n);

}

assign3(assign3 obj1){//copy constructor

System.out.println("copying object using constructor-");

roll=obj1.roll;

name=obj1.name;

}

void display(){

System.out.println(roll+" "+name);

}

public static void main(String[] argv){

int rollno;

String name;

System.out.println("Rollno Name");

assign3 s1=new assign3(1,"raj");

assign3 s2=new assign3(2,"abc");

assign3 s3=new assign3(3,"def");

assign3 s4=new assign3(4,"ghi");

assign3 s5=new assign3(5,"jkl");

assign3 s6=new assign3(s5); s6.display();

assign3 s7=new assign3();

System.out.println("copying object without using constructor-");

s7.roll=s5.roll;

s7.name=s5.name;

s7.display();

}

}

**ASSIGNMENT 4**

# Program to create a class complex with necessary operator overloading and type conversion such as integer to complex, complex to double.

CODE:

//type conversion

class numbers{

int r,i;

numbers(int real,int img){

this.r=real;

this.i=img;

}

numbers plus(numbers x){

return new numbers(r+x.r,i+x.i);

}

}

public class assign4 {

public static void main(String[] args){

//byte -> short -> char -> int -> long -> float -> double(automatic conversion)

//double -> float -> long -> int -> char -> short -> byte(manual conversion)

int num1=1000;double dl=100.23;float fl=(float)200.45;

float f=num1;

double d=num1;

System.out.println("Conversion of int to float-");

System.out.println("int= "+num1+" float= "+f);

System.out.println("Conversion of int to double-");

System.out.println("int= "+num1+" double= "+d);

dl=fl;

System.out.println("Conversion of float to double-");

System.out.println("float= "+f+" double= "+d);

byte bt=2;

int num2=bt;

f=bt;

d=bt;

System.out.println("Conversion of byte to int-");

System.out.println("byte= "+bt+" int= "+num2);

System.out.println("Conversion of byte to float-");

System.out.println("byte= "+bt+" int= "+f);

System.out.println("Conversion of byte to double-");

System.out.println("byte= "+bt+" int= "+d);

bt=(byte)num1;

System.out.println("Conversion of int to byte-");

System.out.println("int= "+num1+" byte= "+bt);

bt=(byte)d;

System.out.println("Conversion of double to byte-");

System.out.println("double= "+d+" byte= "+bt);

numbers obj1=new numbers(10,5);

numbers obj2=new numbers(1,2);

numbers sum=obj1.plus(obj2);

System.out.println("\nsum of two nums is: "+sum.r+" + i"+sum.i);

}

}

**Output🡪**

Conversion of int to float-

int= 1000 float= 1000.0

Conversion of int to double-

int= 1000 double= 1000.0

Conversion of float to double-

float= 1000.0 double= 1000.0

Conversion of byte to int-

byte= 2 int= 2

Conversion of byte to float-

byte= 2 int= 2.0

Conversion of byte to double-

byte= 2 int= 2.0

Conversion of int to byte-

int= 1000 byte= -24

Conversion of double to byte-

double= 2.0 byte= 2

sum of two nums is: 11 + i7

**ASSIGNMENT 5**

#Program that randomly generates complex numbers and write two numbers per line in a file along with an operator(+,-,\*,/) .The numbers are written to file in the format (a+ib).

CODE:

import java.util.Random;

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

public class assign5 {

public static void main(String[] args) {

Random rand = new Random(); // instance of random class

int random1 = rand.nextInt(1000); // generates a random no. less than 1000

int random2 = rand.nextInt(1000);

int random3 = rand.nextInt(1000);

int random4 = rand.nextInt(1000);

System.out.println();

System.out.println("Random Number 1 : " + random1);

System.out.println("Random Number 2(img) : " + random2);

System.out.println("Random Number 3 : " + random3);

System.out.println("Random Number 4(img) : " + random4);

System.out.println();

try{

File myfile=new File("JAVA LAB 5.txt");

if (myfile.createNewFile()){

System.out.println("File created : "+myfile.getName());

}

else {

System.out.println("File already exists");

}

}

catch(IOException e){

System.out.println("An error occurred");

e.printStackTrace();

}

try{

FileWriter mywriter=new FileWriter("JAVA LAB 5.txt");

Complex a = new Complex(random1, random2);

Complex b = new Complex(random3, random4);

mywriter.write("The numbers generated are :");

mywriter.write(" "+random1 +" +"+ random2+"(i)\n");

mywriter.write(" "+random3 +" +"+ random4+"(i)\n\n");

System.out.print("The Sum of complex number " + "a " + "and" + " b is : ");

Complex d = a.sum(b);

d.display();

System.out.println();

mywriter.write("The Sum of complex number " + "a " + "and" + " b is : "+d.a+" +"+d.b+"(i)\n");

System.out.print("The subtraction of complex number " + "a " + "and" + " b is : ");

Complex e = a.subtract(b);

e.display();

System.out.println();

mywriter.write("The subtraction of complex number " + "a " + "and" + " b is : "+e.a+" +"+e.b+"(i)\n");

System.out.print("The multiplication of complex number " + "a " + "and" + " b is : ");

Complex m = a.multi(b);

m.display();

System.out.println();

mywriter.write("The multiplication of complex number " + "a " + "and" + " b is : "+m.a+" +"+m.b+"(i)\n");

System.out.print("The division of complex number " + "a " + "and" + " b is : ");

Complex z = a.division(b);

z.display();

System.out.println();

mywriter.write("The division of complex number " + "a " + "and" + " b is : "+z.a+" +"+z.b+"(i)\n");

mywriter.close();

System.out.println("Successfully wrote to the file");

}

catch (IOException e){

System.out.println("An error occured");

e.printStackTrace();

}

}

}

class Complex {

double a, b;

Complex() {

this.a = 0;

this.b = 0;

}

Complex(double x, double y) {

this.a = x;

this.b = y;

}

public Complex sum(Complex y) {

double m, n;

m = a + y.a;

n = b + y.b;

return (new Complex(m, n));

}

public Complex subtract(Complex y) {

double m, n;

m = a - y.a;

n = b - y.b;

return new Complex(m, n);

}

public Complex multi(Complex y) {

double m, n;

m = (a \* y.a - b \* y.b);

n = (a \* y.b + y.a \* b);

return new Complex(m, n);

}

public Complex division(Complex y) {

double m, n, deno;

deno = (y.a \* y.a + y.b \* y.b);

m = (a \* y.a + b \* y.b) / deno;

n = (y.a \* b - a \* y.b) / deno;

return new Complex(m, n);

}

public void display() {

if (b < 0)

System.out.print(a + " - " + (-1 \* b) + "i");

else

System.out.print(a + " + " + b + "i");

}

}

**Output🡪**

Random Number 1 : 129

Random Number 2(img) : 702

Random Number 3 : 532

Random Number 4(img) : 146

File already exists

The Sum of complex number a and b is : 661.0 + 848.0i

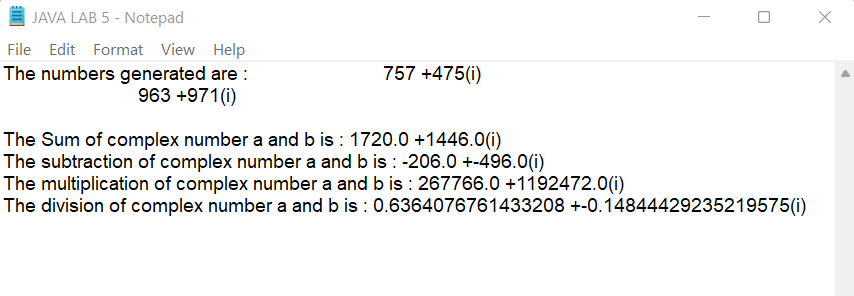
The subtraction of complex number a and b is : -403.0 + 556.0i

The multiplication of complex number a and b is : -33864.0 + 392298.0i

The division of complex number a and b is : 0.5622658868370901 + 1.1652428205296708i

Successfully wrote to the file

**Txt file🡪**



**ASSIGNMENT 6**

# Program to illustrate inheritance.

CODE:

// A->B-> C (Multilevel inheritance)

// A-> D (single inheritance)

// A->D and A-> E (Hierarchical Inheritance)

// I->F and J->F (Multiple Inheritance)

public class ASSIGN6A {

public static void main(String args[]) {

\

System.out.println("Multilevel inheritance");

C obj\_c = new C(); // Multilevel inheritannce

obj\_c.method\_A(); // calling grand super class method

obj\_c.method\_B(); // calling super class method

obj\_c.method\_C(); // calling local function

System.out.println("\nSingle inheritance");

D obj\_D=new D(); // Single inheritance

obj\_D.method\_A(); // Calling super function

obj\_D.method\_D(); // Calling local function

System.out.println("\nHierarchical inheritance");

E obj\_E=new E();

obj\_E.method\_A(); //calling super function of class E

obj\_E.method\_E(); // calling local function of class E

obj\_D.method\_A(); // Calling super function of class D

obj\_D.method\_D(); // Calling local function of class D

System.out.println("\nMultiple inheritance");

F obj\_F=new F();

obj\_F.method\_I(); // Calling interface I's function

obj\_F.method\_J(); // Calling interface J's function

}

}

class A{

public void method\_A() {

System.out.println("This is Class A");

}

}

class B extends A{

public void method\_B() {

System.out.println("This is Class B");

}

}

class C extends B{

public void method\_C() {

System.out.println("This is Class C");

}

}

class D extends A{

public void method\_D() {

System.out.println("This is Class D");

}

}

class E extends A{

public void method\_E() {

System.out.println("This is Class E");

}

}

interface I{

void method\_I();//abstract method

}

interface J{

void method\_J();

}

class F implements I,J{

public void method\_I(){//abstract method implementation

System.out.println("This is Interface I");

}

public void method\_J(){

System.out.println("This is Interface J");

}

}

**Output🡪**

Multilevel inheritance

This is Class A

This is Class B

This is Class C

Single inheritance

This is Class A

This is Class D

Hierarchical inheritance

This is Class A

This is Class E

This is Class A

This is Class D

Multiple inheritance

This is Interface I

This is Interface J

**ASSIGNMENT 7**

# Program to demonstrate the concept of polymorphism.

CODE:

class poly1 {

//compile time polymorphism using function overloading

static int Multiply(int a, int b)

{

return a \* b;

}

static double Multiply(double a, double b)

{

return a \* b;

}

void display(){

System.out.println("This is poly1 class!");

}

}

//Runtime polymorphism

class poly2 extends poly1{

void display(){

System.out.println("This is poly2 class!");

}

}

class poly3 extends poly1{

void display(){

System.out.println("This is poly3 class!");

}

}

class assign7 {

public static void main(String[] args)

{

System.out.println(poly1.Multiply(3, 4));

System.out.println(poly1.Multiply(4.5, 7.3));

poly1 a=new poly2();

poly2 b=new poly2();

a.display();

b.display();

a=new poly3();

a.display();

}

}

**Output🡪**

12

32.85

This is poly2 class!

This is poly2 class!

This is poly3 class!

**ASSIGNMENT 8**

# Program to illustrate Method Overriding.

CODE:

class parent {

void display()

{

System.out.println("display() function of Parent class");

}

}

// Inherited class

class child extends parent {

void display()

{

System.out.println("display() function of Child class");

}

}

// Driver class

class Assign8{

public static void main(String[] args)

{

System.out.println("Raj kumar | 20124083");

parent obj1 = new parent();

obj1.display();

parent obj2 = new child();

obj2.display();

}

}

**Output🡪**

Raj kumar | 20124083

display() function of Parent class

display() function of Child class

**ASSIGNMENT 9**

# Program to illustrate MouseAdapter and KeyAdapter Classes.

CODE:

1.KEY ADAPTER-

import java.awt.\*;

import java.awt.event.\*;

//keyadapter

public class assign9A {

Label label;

public assign9A() {

Frame frame = new Frame();

TextField textField = new TextField();

frame.add(textField, BorderLayout.NORTH);

label = new Label();

frame.add(label, BorderLayout.CENTER);

frame.setSize(450, 400);

frame.setVisible(true);

frame.addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent we){

System.exit(0);

}

});

textField.addKeyListener(new KeyAdapter() {

public void keyTyped(KeyEvent ke){

char keyChar = ke.getKeyChar();

if (keyChar == 'a') {

System.out.println("You typed 'a'");

}

}

public void keyPressed(KeyEvent ke) {

int keyCode = ke.getKeyCode();

if (keyCode == 66) {

System.out.println("You Typed b");

}

}

public void keyReleased(KeyEvent ke){

int keyCode = ke.getKeyCode();

if (keyCode == 67){

System.out.println("You Typed c");

}

}

});

}

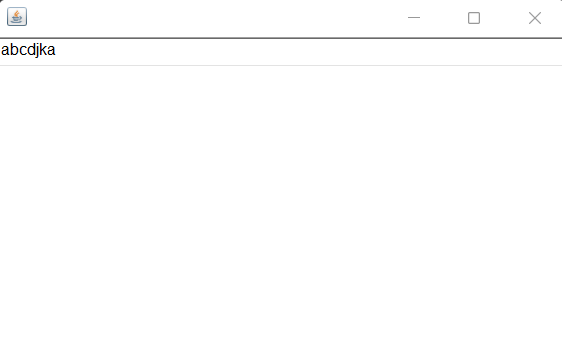
public static void main(String[] args) {

new assign9A();

}

}

Output🡪

****

Terminal:

You typed 'a'

You typed 'a'

You Typed b

You Typed c

You typed 'a'

2.MOUSE ADAPTER-

import java.awt.\*;

import java.awt.event.\*;

//mouse adapter

class assign9B {

public static void main(String args[]) {

Frame frame = new Frame("assign9B");

Button button1 = new Button("PRESS");

Panel p = new Panel();

p.add(button1);

frame.add(button1, BorderLayout.NORTH);

frame.pack();

frame.setVisible(true);

button1.addMouseListener(new MouseAdapter() {

public void mousePressed(MouseEvent me) {

System.out.println("mousePressed " + me.getX());

}

public void mouseReleased(MouseEvent me) {

System.out.println("mouseReleased " + me.getX());

}

}

);

frame.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

System.exit(0);

}

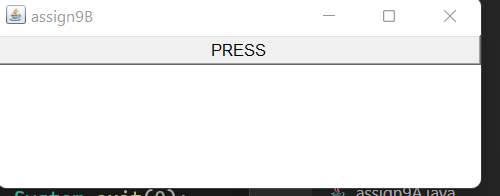
}

);

}

}

**Output🡪**

****

Terminal:

mousePressed 257

mouseReleased 257

mousePressed 257

mouseReleased 257

mousePressed 257

mouseReleased 257

mousePressed 257

mouseReleased 257

mousePressed 257

mouseReleased 257

**ASSIGNMENT 10**

# Program to illustrate Array Manipulation?

CODE:

import java.util.Scanner;

import java.util.Arrays;

class Arrayoperations

{

static void display(int arr[])

{

for(int i:arr)

{

System.out.println(i);

}

}

static int[] newarray()

{

return new int[]{8,7,6,5};

}

}

public class assign10

{ //\*-array-\*

//insertion

//traversal

//returning array

//passing array

//character array

//string array

//coping of one array to another

public static void main(String[] args)

{

//single dimentioanl array

int arr[]=new int[5];//declaration and installation of array of size 5;

Arrayoperations obj=new Arrayoperations();

Scanner inputarr=new Scanner(System.in);

//insertion

System.out.println("# Insertion:");

System.out.println(" Enter 5 Array elements:");

for(int i=0;i<arr.length;i++)

{

arr[i]=inputarr.nextInt();

}

//traversal

System.out.println("# Array traversal:");

for(int i:arr)

{

System.out.println(i);

}

//returning array

int arr2[]=obj.newarray();

System.out.println("\n# Returned array from newarray() function");

System.out.println("Array2" + Arrays.toString(arr2));

//passing array

System.out.println("\n# Passing array to display() function");

System.out.println(" output of display() function:");

obj.display(arr);

//character array

System.out.println("\n# Character Array:");

System.out.println(" Enter elements to character array");

char chararr[]=inputarr.next().toCharArray();

System.out.println("character array" + Arrays.toString(chararr));

inputarr.nextLine();//clear input buffer

//string array

String[] stringarr=new String[5];

System.out.println("\n# Strings Array:");

System.out.println(" Enter 5 strings");

for (int i = 0; i < stringarr.length; i++)

{

stringarr[i] = inputarr.nextLine();

}

System.out.println("string array" + Arrays.toString(stringarr));

//multidimentional array:

System.out.println("\n# Multidimensional Array:");

System.out.println(" Enter element to 3\*3 multidimentional array");

int arr3[][]=new int[3][3];

//insertion

for(int i=0;i<arr3.length;i++){

for(int j=0;j <arr3[0].length ;j++){

arr3[i][j]=inputarr.nextInt();

}

}

//traversal

for(int i=0;i<arr3.length;i++){

for(int j=0;j < arr3[0].length ;j++){

System.out.println(arr3[i][j]);

}

}

//coping of one array to another

System.out.println("# Copying Array");

System.out.println(" Enter 5 elements to copyfrom array");

char copyfrom[]=inputarr.next().toCharArray();

char copyto[]=new char[5];

System.arraycopy(copyfrom, 0, copyto, 0, 5);

System.out.println("copied array" + Arrays.toString(copyto));

}

}

**Output🡪**

# Insertion:

Enter 5 Array elements:

1 2 3 4 5

# Array traversal:

1

2

3

4

5

# Returned array from newarray() function

Array2[8, 7, 6, 5]

# Passing array to display() function

output of display() function:

1

2

3

4

5

# Character Array:

Enter elements to character array

ABC

character array[A, B, C]

# Strings Array:

Enter 5 strings

AAA

BBB

CCC

DDD

EEE

string array[AAA, BBB, CCC, DDD, EEE]

# Multidimensional Array:

Enter element to 3\*3 multidimentional array

1 2 3

4 5 6

7 8 9

1

2

3

4

5

6

7

8

9

# Copying Array

Enter 5 elements to copyfrom array

ABCDE

copied array[A, B, C, D, E]

**ASSIGNMENT 11**

# Program to implement Layout Manager.

CODE:

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.JPanel;

class Border{

Frame f1=new Frame();

Border()

{

Button b1=new Button("North1");

Button b2=new Button("South");

Button b3=new Button("East");

Button b4=new Button("West");

Button b5=new Button("Center");

f1.setLayout(new BorderLayout(30, 10));

f1.add(b1,BorderLayout.NORTH);

f1.add(b2,BorderLayout.SOUTH);

f1.add(b3,BorderLayout.EAST);

f1.add(b4,BorderLayout.WEST);

f1.add(b5,BorderLayout.CENTER);

f1.addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent we) {

System.exit(0);

}

});

f1.setSize(400,400);

f1.setVisible(true);

}

}

class Flow{

Frame f2=new Frame();

Flow(){

Button b1=new Button("1");

Button b2=new Button("2");

Button b3=new Button("3");

Button b4=new Button("4");

Button b5=new Button("5");

f2.setLayout(new FlowLayout(FlowLayout.RIGHT));

f2.add(b1);

f2.add(b2);

f2.add(b3);

f2.add(b4);

f2.add(b5);

f2.setSize(300, 300);

f2.setVisible(true);

}

}

class Gridlayout{

Frame f3=new Frame();

Gridlayout(){

Button b1=new Button("Grid1");

Button b2=new Button("Grid2");

Button b3=new Button("Grid3");

Button b4=new Button("Grid4");

Button b5=new Button("Grid5");

Button b6=new Button("Grid6");

f3.setLayout(new GridLayout(3,2));

f3.add(b1);

f3.add(b2);

f3.add(b3);

f3.add(b4);

f3.add(b5);

f3.add(b6);

f3.setSize(300,300);

f3.setVisible(true);

f3.addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent we) {

System.exit(0);

}

});

}

}

class gridbaglayout

{

Frame f4=new Frame("Gridbaglayout");

gridbaglayout(){

f4.setLayout(new GridBagLayout());

GridBagConstraints c=new GridBagConstraints();

c.gridx=0;

c.gridy=0;

Checkbox c1=new Checkbox("c1");

c1.setBounds(10,10,10,10);

f4.add(c1,c);

c.gridx=0;

c.gridy=1;

Checkbox c2=new Checkbox("c2");

c2.setBounds(10,10,10,10);

f4.add(c2,c);

c.gridx=0;

c.gridy=2;

Checkbox c3=new Checkbox("c3");

c3.setBounds(10,10,10,10);

f4.add(c3,c);

f4.setSize(300,300);

f4.setVisible(true);

}

}

class cardlayout implements ActionListener {

Frame f5=new Frame();

JPanel pane1, pane2, cardPane;

CardLayout card;

cardlayout(){

f5.setLayout(new CardLayout());

pane1 = new JPanel();

pane2 = new JPanel();

cardPane = new JPanel();

Button b1=new Button("next->");

b1.addActionListener(this);

pane1.add(b1);

Button b2=new Button("<-back");

b2.addActionListener(this);

pane2.add(b2);

card = new CardLayout();

cardPane.setLayout(card);

cardPane.add(pane1, "First Pane");

cardPane.add(pane2, "Second Pane");

f5.add(cardPane);

f5.setVisible(true);

f5.setSize(300,300);

}

public void actionPerformed(ActionEvent e) {

card.next(cardPane);

}

}

public class assign11\_layoutmanager {

public static void main(String[] args){

new Border();

new Flow();

new Gridlayout();

new gridbaglayout();

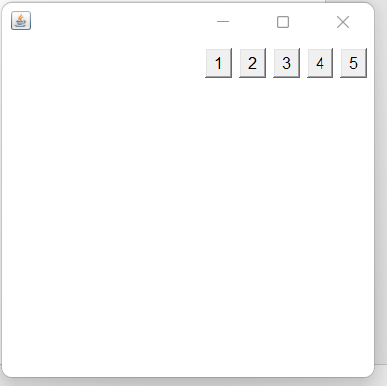
new cardlayout();

}

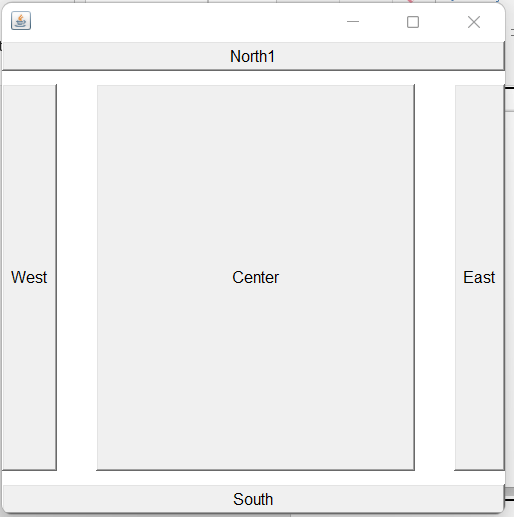
}

**Output🡪**

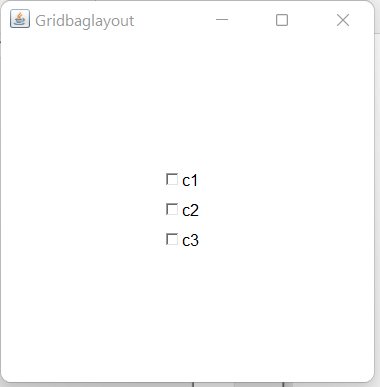
**Flow layout Grid layout**

****

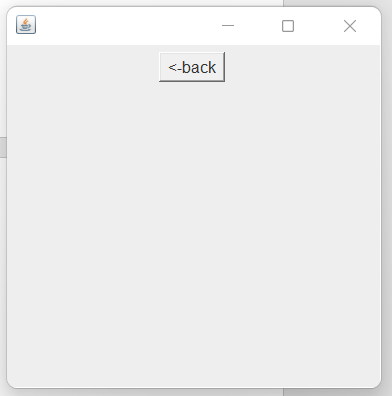
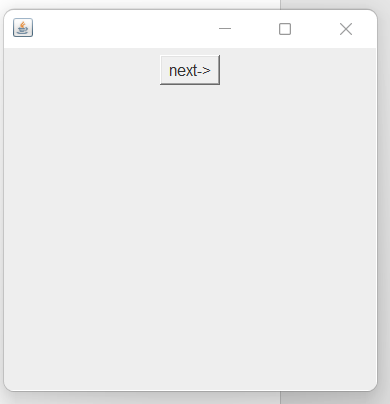
**Border layout**

****

**Gridbag layout**

****

**Card layout**

****