**Assignment No: 2**

1. Write a program to create an interface called area which has a float field PI=3.14f and a method called area(float x,float y) to calculate the area. Implement this interface into 2 classes circle and square and display the results using the class main

interface Area

{

float Pi=3.14f;

float area(float x,float y);

}

class Circle implements Area{

public float area(float x,float y)

{

return(Pi\*x\*x);

}

}

class Square implements Area{

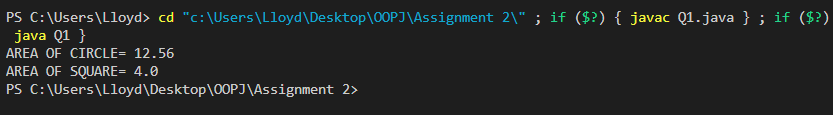
public float area(float x,float y)

{

return(x\*x);

}

}

****

1. Write a program to create a class Student which has an integer variable rollno and has two

methos getNum() and putNum(). Create a class Test which extends Student and has two floating point variables part1 and part2 which are marks. This class has two methods getMarks() and putMarks(). Create an interface called sports which has one floating point variable called sportMarks set to some value, it also has a method to print this value. Create another class called result, which extends class test and implements sports. It has a member total which is the total marks of part1+ppart2+sportMarks. It has two method, to display sports marks, and to display rollno, marks part1 and, part2.

class student{

int roll;

void getNum(int x)

{

roll=x;

}

void putnum()

{

System.out.println("ROLL NUMBER ="+roll);

}

}

class test extends student

{

float put1,put2;

void getmarks(float x,float y)

{

put1=x;

put2=y;

}

void putmarks()

{

System.out.println("P1 ="+put1);

System.out.println("P2 ="+put2);

}

}

interface sports{

float sportMarks=100;

void display();

}

class result extends test implements sports{

int total;

public void display()

{

System.out.println("SPORTS MARKS ="+sportMarks);

}

void display2()

{

float x=put1+put2;

System.out.println("Total ="+x);

}

}

class Q2{

public static void main(String A[])

{

result X=new result();

X.getNum(25);

X.getmarks(100,98);

X.putnum();

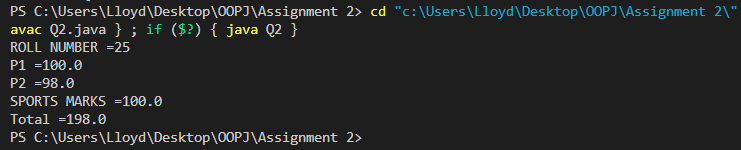
X.putmarks();

X.display();

X.display2();

}

}



1. Write a program to create an interface IN1 and it has a integer variable A=10 and a display method to display the value A. Write another interface IN2 which has an integer b=10 and a method display to display the value of b and another method show. Write a class test Class which implements interface1, write another class testClass2 which implements both the interfaces

interface IN1{

int A=10;

void display1();

}

interface IN2{

int B=10;

void display2();

}

class TestCLASS1 implements IN1{

public void display1()

{

System.out.println("A="+A);

}

}

class TestCLASS2 implements IN1,IN2{

public void display1()

{

System.out.println("A="+A);

}

public void display2()

{

System.out.println("B="+B);

}

}

class Q3{

public static void main(String A[])

{

TestCLASS1 T1=new TestCLASS1();

TestCLASS2 T2=new TestCLASS2();

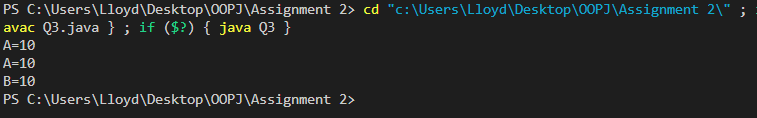
T1.display1();

T2.display1();

T2.display2();

}

}



1. Write a program to create an abstract class animal which has methods eat() and shelter() and an abstract method lifespan() write a child class to implement the abstract methods

abstract class animal{

void eat()

{

System.out.println("INSIDE EAT()");

}

void shelter(){

System.out.println("INSIDE SHELTER()");

}

abstract void lifespan();

}

class child extends animal{

void lifespan()

{

System.out.println("INSIDE CHILD()");

}

}

class Q4{

public static void main(String A[])

{

child C=new child();

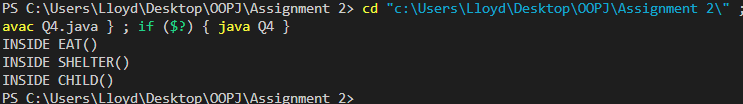
C.eat();

C.shelter();

C.lifespan();

}

}



**Assignment No: 4**

1. Write a java program to implement BorderLayout

import java.awt.\*;

import javax.swing.\*;

public class Main1 {

static void addComponentsToPane(Container pane) {

JButton button = new JButton("TOP");

pane.add(button, BorderLayout.NORTH);

button = new JButton("CENTER");

button.setPreferredSize(new Dimension(200, 100));

pane.add(button, BorderLayout.CENTER);

button = new JButton("LEFT");

pane.add(button, BorderLayout.WEST);

button = new JButton("BOTTOM");

pane.add(button, BorderLayout.SOUTH);

button = new JButton("RIGHT");

pane.add(button, BorderLayout.EAST);

}

private static void GUI() {

JFrame frame = new JFrame("BorderLayout");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

addComponentsToPane(frame.getContentPane());

frame.pack();

frame.setVisible(true);

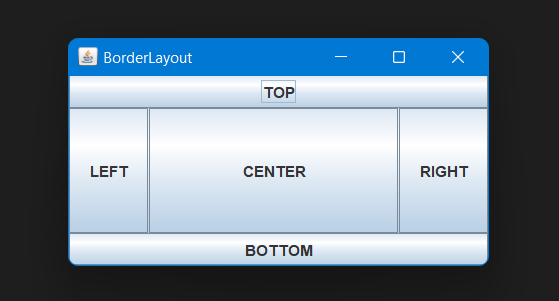
}

public static void main(String[] args) {

GUI();

}

}



1. Write a java program to implement FlowLayout

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class Main2 extends JFrame {

JRadioButton RtoLbutton;

JRadioButton LtoRbutton;

FlowLayout experimentLayout = new FlowLayout();

final String RtoL = "RIGHT TO LEFT";

final String LtoR = "LEFT TO RIGHT";

JButton applyButton = new JButton("APPLY COMPONENT ORIENTATION");

Main2(String name) {

super(name);

}

public void addComponentsToPane(final Container pane) {

JPanel panel = new JPanel();

panel.setLayout(experimentLayout);

experimentLayout.setAlignment(FlowLayout.TRAILING);

JPanel controls = new JPanel();

controls.setLayout(new FlowLayout());

LtoRbutton = new JRadioButton(LtoR);

LtoRbutton.setActionCommand(LtoR);

LtoRbutton.setSelected(true);

RtoLbutton = new JRadioButton(RtoL);

RtoLbutton.setActionCommand(RtoL);

panel.add(new JButton("BUTTON 1"));

panel.add(new JButton("BUTTON 2"));

panel.add(new JButton("BUTTON 3"));

panel.add(new JButton("BUTTON 4"));

panel.add(new JButton("BUTTON 5"));

ButtonGroup group = new ButtonGroup();

group.add(LtoRbutton);

group.add(RtoLbutton);

controls.add(LtoRbutton);

controls.add(RtoLbutton);

controls.add(applyButton);

applyButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String command = group.getSelection().getActionCommand();

if (command.equals("LEFT TO RIGHT")) {

panel.setComponentOrientation(

ComponentOrientation.LEFT\_TO\_RIGHT);

} else {

panel.setComponentOrientation(

ComponentOrientation.RIGHT\_TO\_LEFT);

}

panel.validate();

panel.repaint();

}

});

pane.add(panel, BorderLayout.CENTER);

pane.add(controls, BorderLayout.SOUTH);

;

}

static void GUI() {

Main2 frame = new Main2("FlowLayout");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.addComponentsToPane(frame.getContentPane());

frame.pack();

frame.setVisible(true);

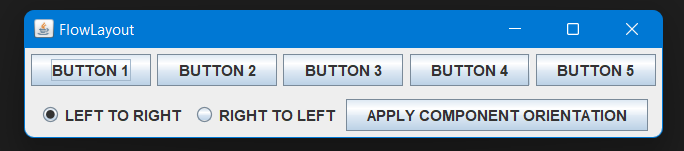
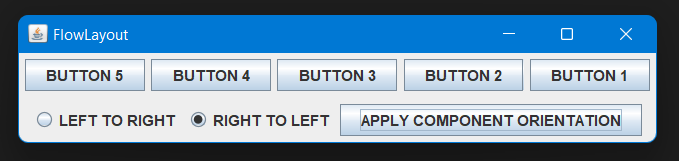
}

public static void main(String[] args) {

GUI();

}

}



1. Write a java program to implement GridLayout

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class Main3 extends JFrame {

String gapList[] = { "0", "5", "10", "15", "20" };

int maxGap = 20;

JComboBox horizontalGapComboBox;

JComboBox verticalGapComboBox;

JButton applyButton = new JButton("APPLY GAPS");

GridLayout experimentLayout = new GridLayout(0, 2);

Main3(String name) {

super(name);

setResizable(false);

}

public void initGaps() {

horizontalGapComboBox = new JComboBox(gapList);

verticalGapComboBox = new JComboBox(gapList);

}

public void addComponentsToPane(final Container pane) {

initGaps();

JPanel panel = new JPanel();

panel.setLayout(experimentLayout);

JPanel controls = new JPanel();

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

JButton b = new JButton("Fake button");

Dimension buttonSize = b.getPreferredSize();

panel.setPreferredSize(new Dimension((int) (buttonSize.getWidth() \* 2.5) + maxGap,

(int) (buttonSize.getHeight() \* 3.5) + maxGap \* 2));

panel.add(new JButton("BUTTON 1"));

panel.add(new JButton("BUTTON 2"));

panel.add(new JButton("BUTTON 3"));

panel.add(new JButton("BUTTON 4"));

panel.add(new JButton("BUTTON 5"));

controls.add(new Label("HORIZONTAL GAP:"));

controls.add(new Label("VERTICAL GAP:"));

controls.add(new Label(" "));

controls.add(horizontalGapComboBox);

controls.add(verticalGapComboBox);

controls.add(applyButton);

applyButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String horizontalGap = (String) horizontalGapComboBox.getSelectedItem();

String verticalGap = (String) verticalGapComboBox.getSelectedItem();

experimentLayout.setHgap(Integer.parseInt(horizontalGap));

experimentLayout.setVgap(Integer.parseInt(verticalGap));

experimentLayout.layoutContainer(panel);

}

});

pane.add(panel, BorderLayout.NORTH);

pane.add(new JSeparator(), BorderLayout.CENTER);

pane.add(controls, BorderLayout.SOUTH);

}

static void GUI() {

Main3 frame = new Main3("GridLayout");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.addComponentsToPane(frame.getContentPane());

frame.pack();

frame.setVisible(true);

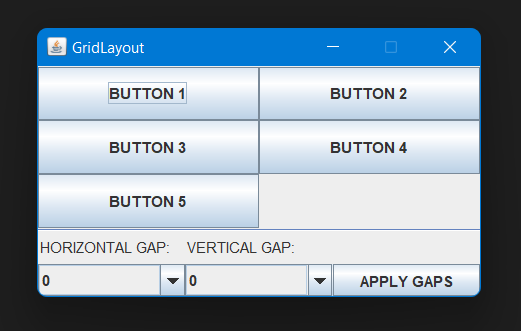
}

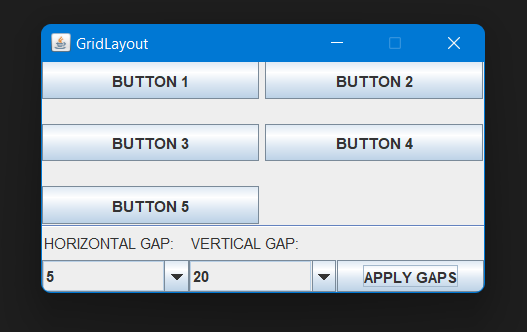
public static void main(String[] args) {

GUI();

}

}





1. Write a java program to implement CardLayout

import java.awt.\*;

import javax.swing.\*;

import java.awt.event.\*;

public class Main4 implements ItemListener {

JPanel cards; // a panel that uses CardLayout

final static String BUTTONPANEL = "CARD WITH BUTTON";

final static String TEXTPANEL = "CARD WITH TEXTFIELD";

public void addComponentToPane(Container pane) {

JPanel comboBoxPane = new JPanel(); // use FlowLayout

String comboBoxItems[] = { BUTTONPANEL, TEXTPANEL };

JComboBox cb = new JComboBox(comboBoxItems);

cb.setEditable(false);

cb.addItemListener(this);

comboBoxPane.add(cb);

JPanel card1 = new JPanel();

card1.add(new JButton("BUTTON 1"));

card1.add(new JButton("BUTTON 2"));

card1.add(new JButton("BUTTON 3"));

JPanel card2 = new JPanel();

card2.add(new JTextField("EBTER HERE ", 20));

cards = new JPanel(new CardLayout());

cards.add(card1, BUTTONPANEL);

cards.add(card2, TEXTPANEL);

pane.add(comboBoxPane, BorderLayout.PAGE\_START);

pane.add(cards, BorderLayout.CENTER);

}

public void itemStateChanged(ItemEvent evt) {

CardLayout cl = (CardLayout) (cards.getLayout());

cl.show(cards, (String) evt.getItem());

}

private static void GUI() {

JFrame frame = new JFrame("CardLayout");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Main4 demo = new Main4();

demo.addComponentToPane(frame.getContentPane());

frame.pack();

frame.setVisible(true);

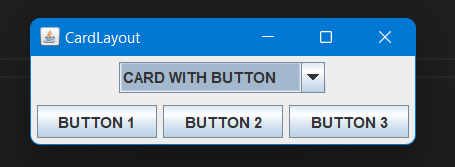
}

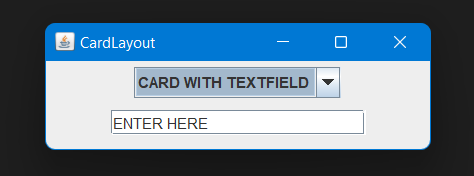
public static void main(String[] args) {

GUI();

}

}





1. Write a java program to implement GridBagLayout

import java.awt.\*;

import javax.swing.\*;

public class Main5 {

public static void addComponentsToPane(Container pane) {

JButton button;

pane.setLayout(new GridBagLayout());

GridBagConstraints c = new GridBagConstraints();

button = new JButton("BUTTON 1");

c.fill = GridBagConstraints.HORIZONTAL;

c.gridx = 0;

c.gridy = 0;

pane.add(button, c);

button = new JButton("BUTTON 2");

c.fill = GridBagConstraints.HORIZONTAL;

c.weightx = 0.5;

c.gridx = 1;

c.gridy = 0;

pane.add(button, c);

button = new JButton("BUTTON 3");

c.fill = GridBagConstraints.HORIZONTAL;

c.weightx = 0.5;

c.gridx = 2;

c.gridy = 0;

pane.add(button, c);

button = new JButton("BUTTON 4");

c.fill = GridBagConstraints.HORIZONTAL;

c.ipady = 40;

c.weightx = 0.0;

c.gridwidth = 3;

c.gridx = 0;

c.gridy = 1;

pane.add(button, c);

button = new JButton("BUTTON 5");

c.fill = GridBagConstraints.HORIZONTAL;

c.ipady = 0;

c.weighty = 1.0;

c.anchor = GridBagConstraints.PAGE\_END;

c.insets = new Insets(10, 0, 0, 0);

c.gridx = 1;

c.gridwidth = 2;

c.gridy = 2;

pane.add(button, c);

}

static void GUI() {

JFrame frame = new JFrame("GridBagLayout");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

addComponentsToPane(frame.getContentPane());

frame.pack();

frame.setVisible(true);

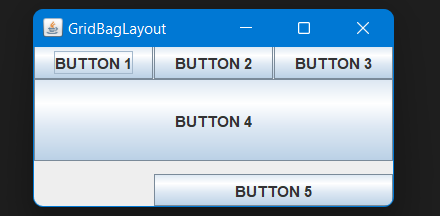
}

public static void main(String[] args) {

GUI();

}

}



**Assignment No: 5**

1. Write a program to run three threads in Java

1. Sort an array & display the result

2. Search for a number in an array

3. Find largest of two number

import java.util.Arrays;

class First extends Thread {

int[] arr;

First(int[] array) {

arr = array;

}

public void run() {

Arrays.sort(arr);

System.out.print("SORTED ARRAY => ");

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

System.out.printf(arr[i] + " ");

}

System.out.println();

}

}

class Second extends Thread {

int[] arr;

int x;

Second(int item, int[] array) {

arr = array;

x = item;

}

public void run() {

int i;

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

if (arr[i] == x) {

System.out.printf("ELEMENT %d FOUND AT POSITION %d\n", x, i + 1);

break;

}

}

if (i == arr.length) {

System.out.printf("ELEMENT %d NOT FOUND\n", x);

}

}

}

class Third extends Thread {

int x;

int y;

Third(int x, int y) {

this.x = x;

this.y = y;

}

public void run() {

if (x > y)

System.out.println(x + " IS THE LARGEST NUMBER");

else

System.out.println(y + " IS THE LARGEST NUMBER");

}

}

class Main1 {

public static void main(String[] args) {

int arr[] = { 1, 4, 98, 2, -1, 59, 23 };

int x = 59;

First first = new First(arr);

Second second = new Second(x, arr);

Third third = new Third(-123, 45);

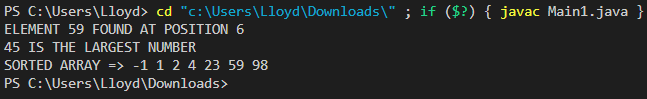
first.start();

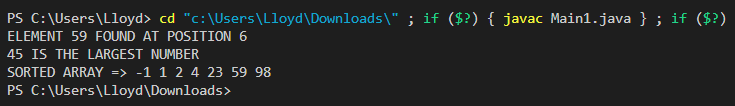
second.start();

third.start();

}

}





1. Write a program to implement runnable interface to create multiple threads First thread will print the first n prime numbers and second thread will print Fibonacci series till n.

class FirstThread implements Runnable {

int n;

FirstThread(int n) {

this.n = n;

}

public void run() {

if (n == 1)

System.out.println(2);

else {

int count = 1, i = 3, j;

System.out.print(2 + " ");

while (count < n) {

for (j = 2; j < i; j++) {

if (i % j == 0)

break;

}

if (i == j) {

count++;

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

}

i++;

}

System.out.println();

}

}

}

class SecondThread implements Runnable {

int n;

SecondThread(int n) {

this.n = n;

}

public void run() {

int a = 0, b = 1, c = a + b;

if (n == 1)

System.out.println(a);

else if (n == 2)

System.out.println(a + " " + b);

else {

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

while (c <= n) {

System.out.print(c + " ");

a = b;

b = c;

c = a + b;

}

}

System.out.println();

}

}

class Main2 {

public static void main(String[] args) {

FirstThread runnableFirstThread = new FirstThread(5);

Thread threadX = new Thread(runnableFirstThread);

SecondThread runnableSecondThread = new SecondThread(38);

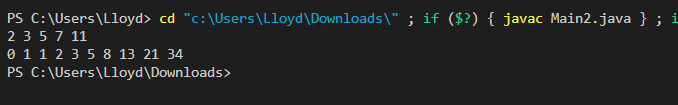
Thread threadY = new Thread(runnableSecondThread);

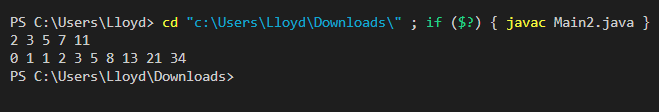
threadX.start();

threadY.start();

}

}





**Assignment No: 3**

1. Implement a package called College which includes the following: • An interface called details which contains a String type variable year, which sets year=TECOMP, methods to set and print name of a student. • A class called TEComp implements this interface. • Two classes: Database and Elejava inherit the above class. • The above two classes have their own methods to set marks of a student. Total marks of Database are calculated as TW+IA+Theory marks. Total marks of Elejava are calculated as IA+Theory marks. Both the classes also implement separate interfaces which have set constant names of faculty as FACULTY1 for Database and FACULTY2 for Java. The source code which has a class Main which includes the main method is in the directory where the package is a subdirectory, has to display the total marks of a student in each subject, with year, roll no of student and faculties.

import college.\*;

public class MAIN {

public static void main(String[] args) {

Database db = new Database();

db.setData("LLOYD ALRICH COSTA", 24, 54, 100, 23);

db.displayData();

System.out.println("");

JAVA\_ELE java = new JAVA\_ELE();

java.setData("RAM DESSAI", 45, 90, 25);

java.displayData();

}

}

package college;

interface Details {

String year = "TECOMP";

void printName();

void setName(String name);

}

public class TECOMP implements Details {

String name;

int rollNo;

public void setName(String name) {

this.name = name;

}

public void printName() {

System.out.println("Name: " + name);

}

public void setRollNo(int rollNo) {

this.rollNo = rollNo;

}

public void printRollNo() {

System.out.println("Roll Number = " + rollNo);

}

}

interface DatabaseFaculty {

String faculty1 = "FACULTY1";

}

package college;

public class Database extends TECOMP implements DatabaseFaculty {

int theory, internal, tw, total;

public void setMarks(int th, int internal, int tw) {

theory = th;

this.internal = internal;

this.tw = tw;

total = theory + internal + tw;

}

public void setData(String name, int rollNo, int theory, int internal, int tw) {

setName(name);

setRollNo(rollNo);

setMarks(theory, internal, tw);

}

public void displayData() {

printName();

printRollNo();

System.out.println("Year = " + year);

System.out.println("Faculty : " + faculty1);

System.out.println("Total = " + total);

}

}

package college;

interface JAVA\_ELE\_Faculty {

String faculty2 = "FACULTY2";

}

public class JAVA\_ELE extends TECOMP implements JAVA\_ELE\_Faculty {

int theory, internal, total;

public void setMarks(int th, int internal) {

theory = th;

this.internal = internal;

total = theory + internal;

}

public void setData(String name, int rollNo, int theory, int internal) {

setName(name);

setRollNo(rollNo);

setMarks(theory, internal);

}

public void displayData() {

printName();

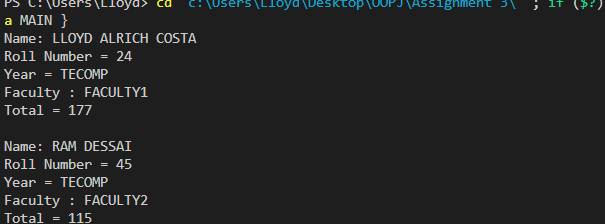
printRollNo();

System.out.println("Year = " + year);

System.out.println("Faculty : " + faculty2);

System.out.println("Total = " + total);

}

}

1. Write a program to create a form that will accept employee id, employee name, employee address and a button to submit using association.

import java.awt.\*;

import java.awt.event.\*;

public class Form {

void myForm(){

Frame f=new Frame("EMPLOYEE FORM");

Label lid=new Label("EMPLOYEE ID");

TextField eid=new TextField(10);

Label lname=new Label("EMPLYEE NAME");

TextField ename=new TextField(10);

Label laddress=new Label("EMPLOYEE ADDRESS");

TextField eaddress=new TextField(10);

Button b=new Button("SUBMIT");

lid.setBounds(10,100,150,20);

eid.setBounds(160,100,150,20);

lname.setBounds(10,120,150,20);

ename.setBounds(160,120,150,20);

laddress.setBounds(10,140,150,20);

eaddress.setBounds(160,140,150,20);

b.setBounds(10,200,100,20);

f.setVisible(true);

f.setLayout(null);

f.setSize(500,400);

f.add(lid);

f.add(eid);

f.add(lname);

f.add(ename);

f.add(eaddress);

f.add(laddress);

f.add(b);

f.addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent e) {

System.exit(0);

}

});

}

public static void main(String[] args) {

Form a = new Form();

a.myForm();

}

}

