***JAVA PRACTICAL LIST QUESTIONS***

***Q1.Design a class complex having real part(x) and an imaginary part(y).Provide methods to perform the following on complex numbers.***

1. ***Add two complex numbers.***
2. ***Multiply two complex numbers.***
3. ***toString() method to display complex numbers in the form: x+iy***

**solution:**

**import** java .io.\*;

**class** complex

{

**private** **int** real,img;

**public** complex()

{

real=0;

img=0;

}

complex(**int** i,**int** j)

{

real=i;

img=j;

}

complex(complex o3)

{

real=o3.real;

img=o3.img;

}

**void** input(**int** p,**int** q)

{

**this**.real=p;

**this**.img=q;

}

**void** display()

{

System.***out***.println(real+"+i"+img);

}

complex add(complex o1)

{

complex temp=**new** complex();

temp.real=**this**.real+o1.real;

temp.img=**this**.img +o1.img;

**return** temp;

}

**public** String toString()

{

**return**(real+"+i"+img);

}

complex multiply(complex o1)

{

complex c=**new** complex();

c.real=**this**.real\*o1.real-**this**.img\*o1.img;

c.img=**this**.img\*o1.real+**this**.real\*o1.img;

**return** c;

}

}

**public** **class** ComplexNumber {

**public** **static** **void** main(String[] args) **throws** IOException{

// **TODO** Auto-generated method stub

complex o1,o2;

**int** x,y;

BufferedReader br=**new** BufferedReader(**new** InputStreamReader(System.***in***));

o1=**new** complex();

System.***out***.println("Enter the real and imaginary part of complex number");

x=Integer.*parseInt*(br.readLine());

y=Integer.*parseInt*(br.readLine());

o1.input(x,y);

o1.display();

o2=**new** complex(4,6);

System.***out***.println("Second complex number");

o2.display();

complex o3;

o3=**new** complex();

o3=o1.add(o2);

System.***out***.println("Sum of two complex number are");

o3.display();

complex o5=**new** complex(o3);

System.***out***.println("Copy constructor");

o5.display();

System.***out***.println("tostring function");

System.***out***.println(o5.toString());

System.***out***.println(o5.add(o1));

complex o6=**new** complex();

o6=o1.multiply(o2);

System.***out***.println("Multiplication of complex numbers");

o6.display();

}

}

**Output:**

Enter the real and imaginary part of complex number

4

9

4+i9

Second complex number

4+i6

Sum of two complex number are

8+i15

Copy constructor

8+i15

tostring function

8+i15

12+i24

Multiplication of complex numbers

-38+i60

***Q2.Create a class TwoDim which contains private members as x and y coordinates in***

***package P1.Define the default constructor, a parameterized constructor and override toString() method to display the coordinates. Now reuse this class and in package P2 create another class ThreeDim, adding a new dimension as z as its private member. Define the constructor for the subclass to override toString() method in the subclass also.Write appropriate methods to show dynamic method to read the data. The main() function should be in a package P.***

**solution:**

**package** p1;

**public** **class** TwoDim {

**private** **int** x,y;

**public** TwoDim()

{

x=y=0;

}

**public** TwoDim(**int** i,**int** y)

{

x=i;

**this**.y=y;

}

**public** String toString()

{

**return**"x and y coordinates"+x+","+y;

}

}

**package** p2;

**import** p1.\*;

**public** **class** ThreeDim **extends** TwoDim {

**private** **int** z;

**public** ThreeDim()

{

z=0;

}

**public** ThreeDim(**int** i,**int** j,**int** k)

{

**super**(i,j);

z=k;

}

**public** String toString() {

**return** **super**.toString()+"z coordinate"+z;

}

}

**package** p;

**import** java.io.\*;

**import** p1.\*;

**import** p2.\*;

**public** **class** coordinatemain {

**public** **static** **void** main(String[] args) **throws** IOException {

// **TODO** Auto-generated method stub

**int** x,y,z;

BufferedReader br=**new** BufferedReader(**new** InputStreamReader(System.***in***));

System.***out***.println("Enter the value of x,y and z");

x=Integer.*parseInt*(br.readLine());

y=Integer.*parseInt*(br.readLine());

z=Integer.*parseInt*(br.readLine());

TwoDim o1= **new** TwoDim ();

TwoDim o2=**new** TwoDim (x,y);

System.***out***.println(o1);

System.***out***.println(o2);

ThreeDim o3=**new** ThreeDim();

ThreeDim o4=**new** ThreeDim(x,y,z);

System.***out***.println(o3);

System.***out***.println(o4);

TwoDim t;

t=o3;

System.***out***.println("By dynamic dispatch");

System.***out***.println(t);

t=o4;

System.***out***.println(t);

}

}

**Output:**

Enter the value of x,y and z

6

7

8

x and y coordinates0,0

x and y coordinates6,7

x and y coordinates0,0z coordinate0

x and y coordinates6,7z coordinate8

By dynamic dispatch

x and y coordinates0,0z coordinate0

x and y coordinates6,7z coordinate8

***Q3. Define an abstract class Shape in package P1.Inherit two more classes: Rectangle in package P2 and Circle in package P3.Write a program to ask the user for the type of shape and this using the concept of dynamic method dispatch, display the area of the appropriate subclass. Also write appropriate methods to read the data. The main() function should not be in any package.***

**Solution:**

**package** P1;

**public** **abstract** **class** Shape {

**protected** **double** area1;

**public** **abstract** **void** area() ;

**public** **void** display()

{

System.***out***.println("Area is undefined");

}

}

**package** P2;

**import** P1.\*;

**public** **class** Rectangle **extends** Shape {

**private** **double** length,breadth;

**public** Rectangle(**double** x,**double** y)

{

length=x;

breadth=y;

}

**public** **void** input(**double** s,**double** r)

{

length=s;

breadth=r;

}

**public** **void** area() {

area1=length\*breadth;

}

**public** **void** display() {

System.***out***.println("Area of Rectangle is:"+area1);

}

}

**package** P3;

**import** P1.\*;

**public** **class** Circle **extends** Shape {

**private** **double** raduis;

**public** Circle(**int** p){

raduis=p;

}

**public** **void** input (**double** r)

{

raduis=r;

}

**public** **void** area() {

area1=3.14\*raduis\*raduis;

}

**public** **void** display() {

System.***out***.println("Area of Circle is:"+area1);

}

}

**import** java.io.\*;

**import** P1.\*;

**import** P2.\*;

**import** P3.\*;

**public** **class** figure {

**public** **static** **void** main(String[] args) **throws** IOException{

// **TODO** Auto-generated method stub

BufferedReader br=**new** BufferedReader(**new** InputStreamReader(System.***in***));

Rectangle r=**new** Rectangle(9,8);

Circle c=**new** Circle(90);

**double** x,y;

String ch1;

**int** x1;

System.***out***.println("1.Rectangle");

System.***out***.println("2.Circle");

System.***out***.println("3.By dynamic dispatch");

**do** {

System.***out***.println("Enter your choice");

x1=Integer.*parseInt*(br.readLine());

**switch**(x1) {

**case** 1: r.area();

r.display();

System.***out***.println("enter the value of x and y");

x=Double.*parseDouble*(br.readLine());

y=Double.*parseDouble*(br.readLine());

r.input(x, y);

r.area();

r.display();

**break**;

**case** 2: c.area();

c.display();

**break**;

**case** 3: System.***out***.println("by using dynamic dispatch:");

Shape s;

s=r;

s.display();

s=c;

s.display();

}

System.***out***.println("Do you want to continue");

ch1=br.readLine();

}**while**(ch1.equals("y"));

}

}

**Output:**

1.Rectangle

2.Circle

3.By dynamic dispatch

Enter your choice

1

Area of Rectangle is:72.0

enter the value of x and y

6

8

Area of Rectangle is:48.0

Do you want to continue

y

Enter your choice

2

Area of Circle is:25434.000000000004

Do you want to continue

y

Enter your choice

3

by using dynamic dispatch:

Area of Rectangle is:48.0

Area of Circle is:25434.000000000004

Do you want to continue

n

***Q4.Create an exception subclass Underage, which prints “Underage Age” along with the age value when an object of Underage class is printed in the catch statement. Write a class exceptionDemo in which the method test() throws Underage exception if the variable age is passed to it as argument is less than 18.Write main() method also to show working of the program.***

**Solution:**

**import** java.io.\*;

**class** Underage **extends** Exception

{

**int** age;

Underage(**int** p)

{

age=p;

}

**public** String toString()

{

**return** "Underage"+age;

}

}

**public** **class** exceptionunderage {

**static** **void** test(**int** x) **throws** Underage

{

**if**(x<18)

**throw** **new** Underage(x);

System.***out***.println("Age is:" +x);

}

**public** **static** **void** main(String[] args) **throws** IOException {

// **TODO** Auto-generated method stub

BufferedReader br=**new** BufferedReader(**new** InputStreamReader(System.***in***));

String ch="y";

**do**

{

System.***out***.println("Enter the age");

**int** x=Integer.*parseInt*(br.readLine());

**try**

{

*test*(x);

}

**catch**(Exception o1)

{

System.***out***.println(o1);

}

System.***out***.println("Do you want to continue");

ch=br.readLine();

}**while**(ch.equals("y"));

System.***out***.println("End of Underage");

}

}

**Output:**

Enter the age

67

Age is:67

Do you want to continue

y

Enter the age

4

Underage4

Do you want to continue

n

End of Underage

***Q5.Write a program to implement stack. Use exception handling to manage underflow and overflow conditions.***

**Solution:**

**import** java.io.\*;

**class** stackexcp **extends** Exception

{

String str;

stackexcp(String a)

{

str=a;

}

**public** String toString()

{

**return** "stack" +str;

}

}

**class** stack

{

**int** a[],size;

**int** tos;

stack(**int** k)

{

size=k;

a=**new** **int**[size];

tos=-1;

}

**void** push(**int** x)

{

**try**

{

**if**(tos==size-1) {

**throw** **new** stackexcp("overflow");

}

**else**

a[++tos]=x;

}

**catch**(stackexcp o1)

{

System.***out***.println(o1);

}

}

**void** pop()

{

**try**

{

**if**(tos==-1) {

**throw** **new** stackexcp("Underflow");

}

**else**

System.***out***.println(a[tos--]);

}

**catch**(stackexcp o1)

{

System.***out***.println(o1);

}

}

**void** display()

{

**for**(**int** i=tos;i>=0;i--)

System.***out***.println(a[i]);

}

}

**public** **class** stackclass {

**public** **static** **void** main(String[] args) **throws** IOException{

// **TODO** Auto-generated method stub

stack o1=**new** stack(4);

**int** x;

String ch1;

BufferedReader br=**new** BufferedReader(**new** InputStreamReader(System.***in***));

**do**

{

System.***out***.println("1.Push operation:");

System.***out***.println("2.Pop operation:");

System.***out***.println("3.Display operation:");

System.***out***.println("Enter your choice:");

**int** ch=Integer.*parseInt*(br.readLine());

**switch**(ch)

{

**case** 1: System.***out***.println("Enter the value to push");

x=Integer.*parseInt*(br.readLine());

o1.push(x);

**break**;

**case** 2: **for**(**int** i=1;i<=6;i++)

o1.pop();

**break**;

**case** 3: o1.display();

}

System.***out***.println("Do you want to continue");

ch1=br.readLine();

}**while**(ch1.equals("y"));

}

}

**Output:**

1.Push operation:

2.Pop operation:

3.Display operation:

Enter your choice:

1

Enter the value to push

45

Do you want to continue

y

1.Push operation:

2.Pop operation:

3.Display operation:

Enter your choice:

1

Enter the value to push

56

Do you want to continue

y

1.Push operation:

2.Pop operation:

3.Display operation:

Enter your choice:

1

Enter the value to push

34

Do you want to continue

y

1.Push operation:

2.Pop operation:

3.Display operation:

Enter your choice:

8

Do you want to continue

y

1.Push operation:

2.Pop operation:

3.Display operation:

Enter your choice:

1

Enter the value to push

89

Do you want to continue

y

1.Push operation:

2.Pop operation:

3.Display operation:

Enter your choice:

1

Enter the value to push

24

stackoverflow

Do you want to continue

y

1.Push operation:

2.Pop operation:

3.Display operation:

Enter your choice:

2

89

34

56

45

stackUnderflow

stackUnderflow

Do you want to continue

n

***Q6.Write a program that copies content of one file to another. Pass the names through command-line argument.***

***Solution:***

**import** java .io.\*;

**public** **class** filecopy {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

FileInputStream f1=**null**;

FileOutputStream f2=**null**;

**int** i;

**for**(**int** i1=0;i1<args.length;i1++)

System.***out***.println("args["+i1+"]"+args[i1]);

**if**(args.length!=2)

{

System.***out***.println("File name out mentioned");

**return**;

}

**try**

{

f1=**new** FileInputStream(args[0]);

f2=**new** FileOutputStream(args[1]);

**do**

{

i=f1.read();

**if**(i!=-1)

f2.write(i);

}**while**(i!=-1);

}

**catch**(FileNotFoundException o1)

{

System.***out***.println(o1);

}

**catch**(IOException o1)

{

System.***out***.println(o1);

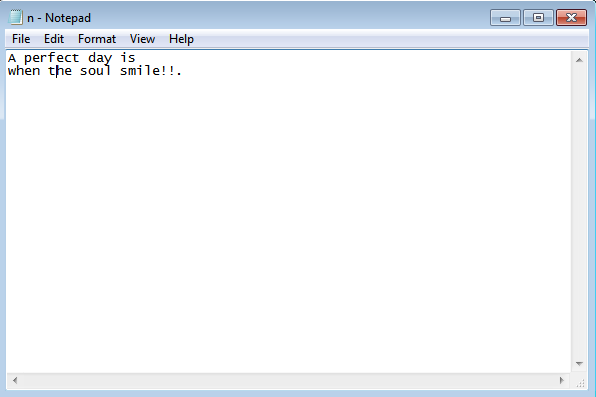
}

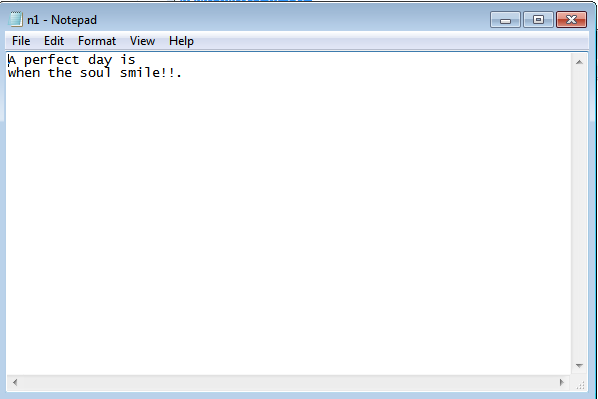
}

**Output:**

args[0]C:\Users\user\eclipse-workspace\filehandling\src\n.txt

args[1]C:\Users\user\eclipse-workspace\filehandling\src\n1.txt

****

****

***Q7.Write a program to read a file and display only those lines that have the first two characters as ‘//’.***

**Solution:**

**import** java.io.\*;

**public** **class** filecontentdisplay {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

FileInputStream f1=**null**;

**int** i1,i;

**char** ch,ch1,ch2;

**if**(args.length!=1)

{

System.***out***.println("File name not mentioned");

**return**;

}

**try**

{

f1=**new** FileInputStream(args[0]);

**do** {

i=f1.read();

**if**(i!=-1)

{

i1=f1.read();

ch=(**char**) i;

ch1=(**char**) i1;

**do**

{

ch2=(**char**)f1.read();

**if**(ch=='/'&& ch1=='/')

System.***out***.print(ch2);

}**while**(ch2!='\n');

}

}**while**(i!=-1);

}

**catch**(FileNotFoundException o1)

{

System.***out***.println(o1);

}

**catch**(IOException o1)

{

System.***out***.println(o1);

}

}

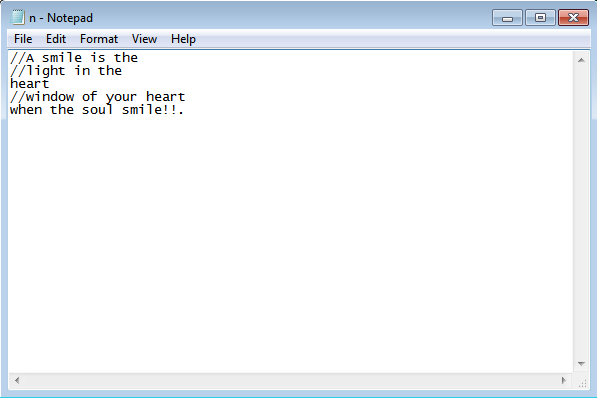
}

**Output:**

A smile is the

light in the

window of your heart

****

**Q8*.******Write a program to create a frame using AWT. Implement mouseClicked(), mouseEntered() and mouseExited() events such that:***

***a) Size of the frame should be tripled when mouse enters it.***

***b) Frame should reduce to its original size when mouse is clicked in it.***

***c) Close the frame when mouse exits it.***

**Solution:**

**import** java.awt.\*;

**import** java.awt.event.\*;

**class** MyFrame **extends** Frame **implements** MouseListener,WindowListener{

String msg ="Hello";

Dimension d1;

MyFrame(String s)

{

**super**(s);

addMouseListener(**this**);

addWindowListener(**this**);

setSize(300,300);

d1=getSize();

}

**public** **void** windowClosed(WindowEvent ae)

{

}

**public** **void** windowAcitvated(WindowEvent ae)

{

}

**public** **void** windowClosing(WindowEvent ae)

{

setVisible(**false**);

}

**public** **void** windowDeactivated(WindowEvent ae)

{

}

**public** **void** windowDeiconified(WindowEvent ae)

{

}

**public** **void** windowIconfied(WindowEvent ae)

{

}

**public** **void** windowOpened(WindowEvent ae)

{

}

**public** **void** mouseClicked(MouseEvent ae1)

{

msg="Mouse Clicked window size="+d1.width+"and"+d1.height;

setSize(d1);

repaint();

}

**public** **void** mousePressed(MouseEvent ae1)

{

msg="Mouse Pressed";

repaint();

}

**public** **void** mouseReleased(MouseEvent ae1)

{

msg="Mouse Released";

repaint();

}

**public** **void** mouseEntered(MouseEvent ae1)

{

Dimension d;

msg="Mouse Entered";

d=getSize();

setSize(d.width\*3,d.height\*3);

repaint();

}

**public** **void** mouseExited(MouseEvent ae1)

{

msg="Mouse Exited";

repaint();

setVisible(**false**);

}

**public** **void** paint(Graphics g)

{

g.drawString(msg,50,100);

}

}

**public** **class** mouseevenhandling {

**public** **static** **void** main(String[] args) {

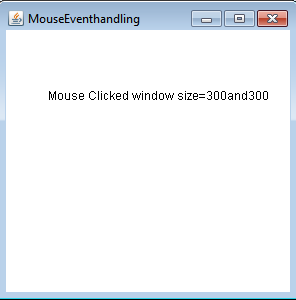
// **TODO** Auto-generated method stub

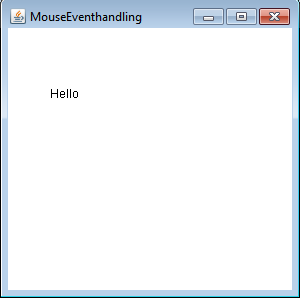
MyFrame o1=**new** MyFrame("MouseEventhandling");

o1.setVisible(**true**);

}

}

**Output:**

****

***Q9.Using AWT,write a program to display a string in frame window with pink color as***

***background.***

**Solution:**

**import** java.awt.\*;

**import** java.awt.event.\*;

**class** myframe2 **extends** Frame **implements** WindowListener

{

String msg;

myframe2(String s,String s1)

{

**super**(s);

setBackground(Color.***pink***);

msg=s1;

addWindowListener(**this**);

repaint();

}

**public** **void** windowClosed(WindowEvent ae)

{

}

**public** **void** windowAcitvated(WindowEvent ae)

{

}

**public** **void** windowClosing(WindowEvent ae)

{

setVisible(**false**);

}

**public** **void** windowDeactivated(WindowEvent ae)

{

}

**public** **void** windowDeiconified(WindowEvent ae)

{

}

**public** **void** windowIconfied(WindowEvent ae)

{

}

**public** **void** windowOpened(WindowEvent ae)

{

}

**public** **void** paint(Graphics g)

{

g.drawString(msg,150,100);

}

}

**public** **class** framebackground {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

myframe2 o2=**new** myframe2("FrameBackground","Hello World");

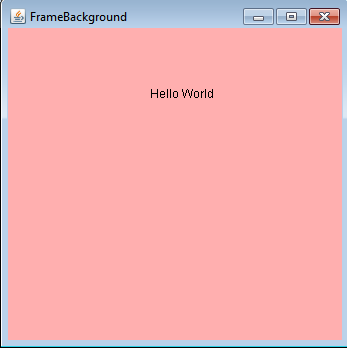
o2.setSize(**new** Dimension(350,350));

o2.setVisible(**true**);

}

}

**Output:**



**Q10.*Using AWT,write a program to create two buttons named “Red” and “Blue”. When a button is pressed the background color should be set to the color named by the button’s label.***

***Solution:***

**import** java.awt.\*;

**import** java.awt.event.\*;

**class** Myframe **extends** Frame **implements** ActionListener

{

Button b1,b2;

Myframe()

{

setLayout(**new** FlowLayout());

b1=**new** Button("Red");

b2=**new** Button("Blue");

add(b1);

add(b2);

b1.addActionListener(**this**);

b2.addActionListener(**this**);

addWindowListener(**new** WindowAdapter(){

**public** **void** windowClosing(WindowEvent we)

{

System.*exit*(0);

}

});

}

**public** **void** actionPerformed(ActionEvent ae)

{

String str=ae.getActionCommand();

**if**(str.equals("Red"))

setBackground(Color.***red***);

**else** **if**(str.equals("Blue"))

setBackground(Color.***blue***);

}

}

**public** **class** Buttoneventhandling {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Myframe o1=**new** Myframe();

o1.setTitle("BACKGROUND");

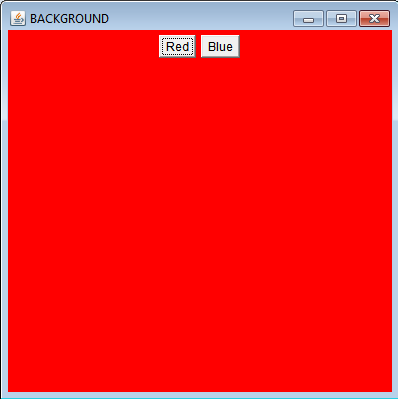
o1.setSize(**new** Dimension(400,400));

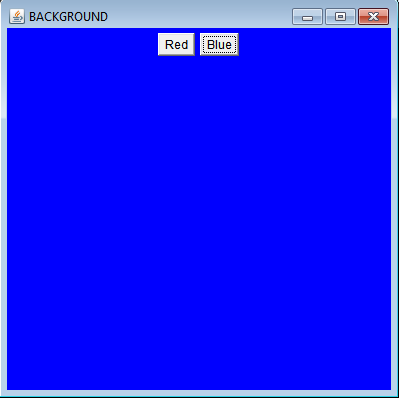
o1.setVisible(**true**);

}

}

**Output:**





**Q11.*Using AWT,write a program using appropriate adapter class to display the message (“Typed character is:<typeCharacter>”) in the frame window when user types any key.***

***Solution:***

**import** java.awt.\*;

**import** java.awt.event.\*;

**class** Myframek **extends** Frame

{

String msg="Typed characters are";

String msg1=" ";

String msg2=" ";

Myframek(String str)

{

**super**(str);

setBackground(Color.***pink***);

setForeground(Color.***blue***);

addKeyListener(**new** MyKeyAdapter(**this**));

addWindowListener(**new** MyWindowAdapter(**this**));

}

**public** **void** paint(Graphics g)

{

g.drawString(msg,100,100);

g.drawString(msg1,100,120);

g.drawString(msg2,100,140);

}

}

**class** MyKeyAdapter **extends** KeyAdapter

{

Myframek o1;

**int** counter=0;

MyKeyAdapter(Myframek o2)

{

o1=o2;

}

**public** **void** keyTyped(KeyEvent ke)

{

counter++;

**if**(counter<=20)

o1.msg+=ke.getKeyChar();

**else** **if**(counter<=40)

o1.msg1+=ke.getKeyChar();

**else**

o1.msg2+=ke.getKeyChar();

o1.repaint();

}

}

**class** MyWindowAdapter **extends** WindowAdapter

{

Myframek o1;

MyWindowAdapter(Myframek o2)

{

o1=o2;

}

**public** **void** windowClosing(WindowEvent we)

{

o1.setVisible(**false**);

}

}

**public** **class** KeyTypedEvent {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Myframek o1=**new** Myframek("KeyTyped");

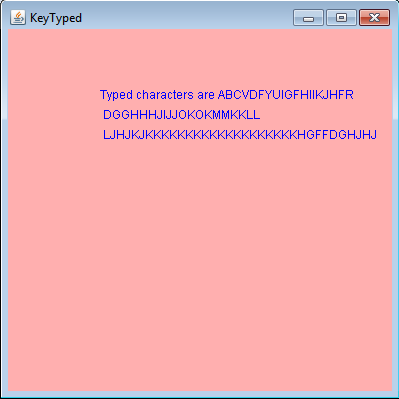
o1.setSize(**new** Dimension(400,400));

o1.setVisible(**true**);

}

}

**Output:**



**Q.12)*Using AWT,write a program to create two buttons labeled ‘A’ and ‘B’.When button ‘A’ is pressed,it displays your personal information(Name,Course,Roll No, College) and when button ‘B’ is pressed,it display your CGPA in previous semester.***

***Solution:***

**import** java.awt.\*;

**import** java.awt.event.\*;

**class** Studentinfo

{

String Name,Course,College;

**int** Rollno;

Double cgpa;

Studentinfo(String name,String course,String college,**int** rollno,**double** cgpa)

{

Name=name;

Course=course;

College=college;

Rollno=rollno;

**this**.cgpa=cgpa;

}

}

**class** Myframe1 **extends** Frame **implements** ActionListener

{

Studentinfo s1=**new** Studentinfo("Neha","Bsc Hons ComputerScience","Shaheed Rajguru College of Applied Sciences for Women",2020331,9.9);

Button b1,b2;

Label l1,l2,l3,l4;

Myframe1()

{

setLayout(**new** FlowLayout());

b1=**new** Button("A");

b2=**new** Button("B");

l1=**new** Label();

l2=**new** Label();

l3=**new** Label();

l4=**new** Label();

add(b1);

add(b2);

add(l1);

add(l2);

add(l3);

add(l4);

b1.addActionListener(**this**);

b2.addActionListener(**this**);

addWindowListener(**new** WindowAdapter(){

**public** **void** windowClosing(WindowEvent we)

{

System.*exit*(0);

}

});

}

**public** **void** actionPerformed(ActionEvent ae)

{

String s=ae.getActionCommand();

l1.setText(" ");

l2.setText(" ");

l3.setText(" ");

l4.setText(" ");

**if**(s.equals("A"))

{

l1.setText("Name-"+s1.Name);

l2.setText("Course-"+s1.Course);

l3.setText("College-"+s1.College);

l4.setText("Rollno-"+s1.Rollno);

}

**else** **if**(s.equals("B"))

l1.setText("Cgpa-"+s1.cgpa);

}

}

**public** **class** Studentbuttonevent {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Myframe1 o1=**new** Myframe1();

o1.setTitle("Student");

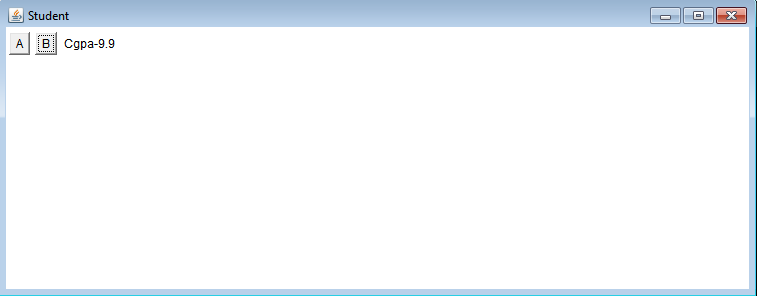
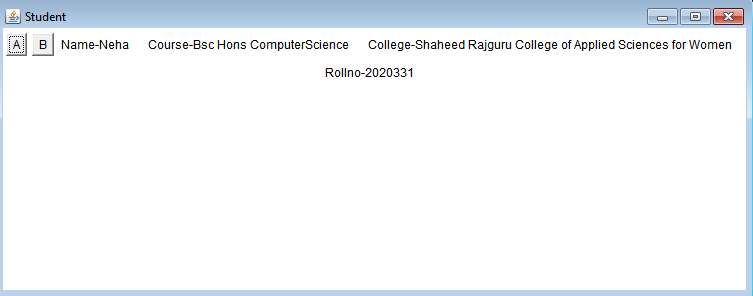
o1.setSize(**new** Dimension(300,300));

o1.setVisible(**true**);

}

}

**Output:**

****

***Q13)Using Swings rewrite the above awt questions.***

\***Q8)using swings.**

**Solution:**

**import** java.awt.\*;

**import** java.awt.event.\*;

**import** javax.swing.\*;

**class** mouseF

{

Dimension d;

JFrame o1;

mouseF()

{

o1=**new** JFrame("MouseEventHandling Swings");

o1.setSize(400,400);

o1.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

d=o1.getSize();

o1.getContentPane().setBackground(Color.***blue***);

o1.addMouseListener(**new** MyMouseAdapter(o1,d));

o1.setVisible(**true**);

}

}

**class** MyMouseAdapter **extends** MouseAdapter

{

JFrame o1;

Dimension d;

MyMouseAdapter(JFrame o2,Dimension d1)

{

o1=o2;

d=d1;

}

**public** **void** mouseEntered(MouseEvent me)

{

Dimension d1;

d1=o1.getSize();

o1.setSize(d1.width\*3,d1.height\*3);

}

**public** **void** mouseClicked(MouseEvent me)

{

o1.setSize(d);

}

**public** **void** mouseExited(MouseEvent me)

{

o1.setVisible(**false**);

}

}

**public** **class** mouseSwing {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

SwingUtilities.*invokeLater*(**new** Runnable()

{

**public** **void** run()

{

**new** mouseF();

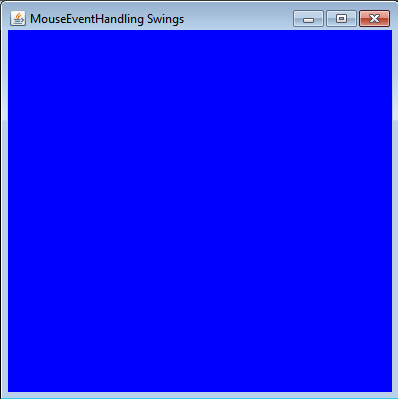
}

});

System.***out***.println("hello");

}

}

**Output:**

***\**Q9)Using swings:**

**Solution:**

**import** java.awt.\*;

**import** java.awt.event.\*;

**import** javax.swing.\*;

**class** panelS **extends** JPanel

{

String s="hello";

**protected** **void** paintComponent(Graphics g)

{

**super**.paintComponent(g);

setBackground(Color.***pink***);

g.drawString(s,100,100);

}

}

**class** frames

{

JFrame o1;

panelS p;

frames()

{

o1=**new** JFrame("frame Swing");

p=**new** panelS();

o1.setSize(400,400);

o1.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

o1.add(p);

o1.setVisible(**true**);

}

}

**public** **class** frameSwing {

**public** **static** **void** main(String[] args)

{

SwingUtilities.*invokeLater*(**new** Runnable()

{

**public** **void** run()

{

**new** frames();

}

});

}

}

**Output:**

****

**\*Q10)Using swings.**

**Solution:**

**import** java.awt.\*;

**import** java.awt.event.\*;

**import** javax.swing.\*;

**class** panelk **extends** JPanel

{

String msg="Typed characters: ";

String msg1=" ";

String msg2=" ";

**protected** **void** paintComponent(Graphics g)

{

**super**.paintComponent(g);

g.drawString(msg,150,150);

g.drawString(msg1,150,160);

g.drawString(msg2,150,170);

}

}

**class** framek **extends** JFrame

{

panelk p;

framek()

{

**super**("key event handling swimgs");

setSize(450,450);

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

p=**new** panelk();

add(p);

addKeyListener(**new** MyKeyAdapter(p));

setVisible(**true**);

}

}

**class** MyKeyAdapter **extends** KeyAdapter

{

**int** counter=0;

panelk o1;

MyKeyAdapter(panelk o2)

{

o1=o2;

}

**public** **void** keyTyped(KeyEvent ke)

{

counter++;

**if**(counter<=20)

o1.msg+=ke.getKeyChar();

**else** **if**(counter<=40)

o1.msg1+=ke.getKeyChar();

**else**

o1.msg2+=ke.getKeyChar();

o1.repaint();

}

}

**public** **class** keyeventhandlingSwings {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

SwingUtilities.*invokeLater*(**new** Runnable()

{

**public** **void** run()

{

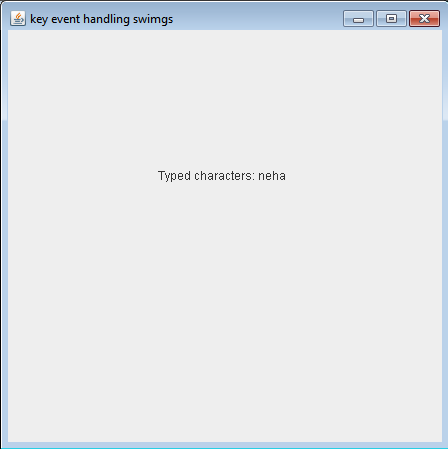
**new** framek();

}

});

}

}

**Output:**

**\*Q11)Using swings.**

**Solution:**

**import** java.awt.\*;

**import** java.awt.event.\*;

**import** javax.swing.\*;

**class** SwingsDemo **implements** ActionListener

{

JButton b1,b2;

JFrame o1;

SwingsDemo()

{

o1=**new** JFrame("Buttonswings");

b1=**new** JButton("Red");

b2=**new** JButton("Blue");

o1.setSize(400,400);

o1.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

o1.setLayout(**new** FlowLayout());

o1.getContentPane().add(b1);

o1.getContentPane().add(b2);

b1.addActionListener(**this**);

b2.addActionListener(**this**);

o1.setVisible(**true**);

}

**public** **void** actionPerformed(ActionEvent ae)

{

String s;

s=ae.getActionCommand();

**if**(s.equals("Red"))

o1.getContentPane().setBackground(Color.***red***);

**else** **if**(s.equals("Blue"))

o1.getContentPane().setBackground(Color.***blue***);

}

}

**public** **class** swingbutton {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

SwingUtilities.*invokeLater*(**new** Runnable()

{

**public** **void** run()

{

**new** SwingsDemo();

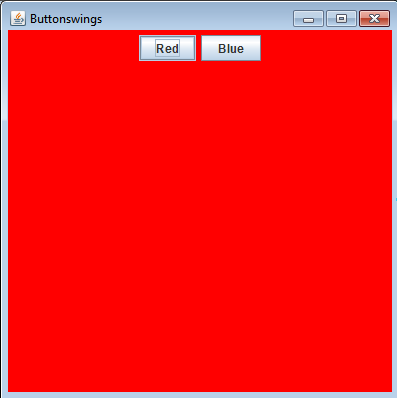
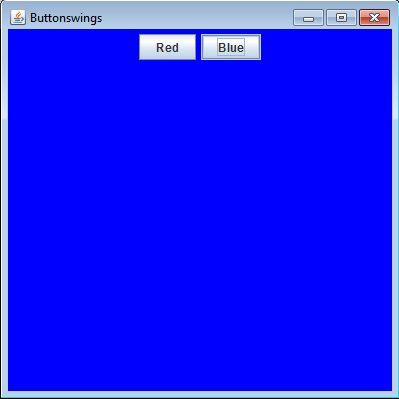
}

});

}

}

**Output:**

****

**\*Q12)Using swings.**

**Solution:**

**import** java.awt.\*;

**import** java.awt.event.\*;

**import** javax.swing.\*;

**class** Student

{

String name,course,college;

**int** Rollno;

**double** cgpa;

Student(String a, String b,String c,**int** r,**double** d)

{

name=a;

course=b;

college=c;

Rollno=r;

cgpa=d;

}

}

**class** panelab **extends** JPanel

{

String msg1=" ";

String msg2=" ";

String msg3=" ";

String msg4=" ";

**protected** **void** paintComponent(Graphics g)

{

**super**.paintComponent(g);

g.drawString(msg1, 150, 200);

g.drawString(msg2, 150, 220);

g.drawString(msg3, 150, 240);

g.drawString(msg4, 150, 260);

}

}

**class** buttonab **extends** JFrame **implements** ActionListener

{

JButton b1,b2;

panelab p1;

// JLabel l;

Student s1=**new** Student("neha","bsc computer science","srcaw",2020331,9.9);

buttonab()

{

setTitle("button a and b");

setSize(400,400);

p1=**new** panelab();

b1=**new** JButton("A");

b2=**new** JButton("B");

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

p1.add(b1);

p1.add(b2);

add(p1);

b1.addActionListener(**this**);

b2.addActionListener(**this**);

setVisible(**true**);

}

**public** **void** actionPerformed(ActionEvent ae)

{

String s=ae.getActionCommand();

**if**(s.equals("A"))

{

p1.msg1="name-"+s1.name;

p1.msg2="course-"+s1.course;

p1.msg3="college-"+s1.college;

p1.msg4="Rollno-"+s1.Rollno;

p1.repaint();

}

**else** **if**(s.equals("B")) {

p1.msg1="cgpa-"+s1.cgpa;

p1.msg2=" ";

p1.msg3=" ";

p1.msg4="";

p1.repaint();

}

}

}

**public** **class** ButtonAandB {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

SwingUtilities.*invokeLater*(**new** Runnable()

{

**public** **void** run()

{

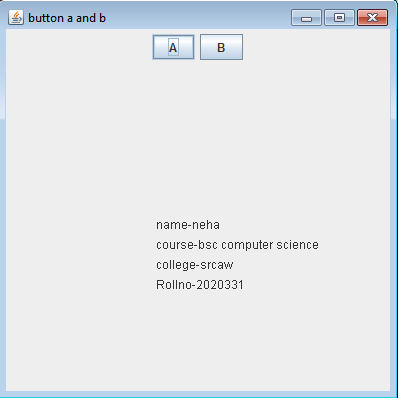
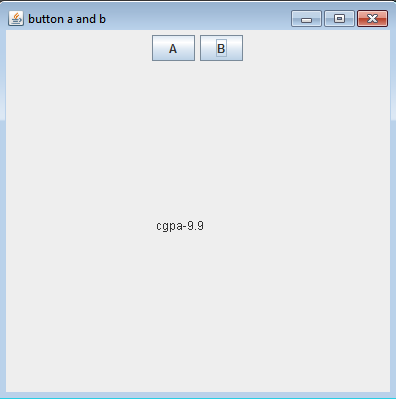
**new** buttonab();

}

});

}

}

**Output:**

**Name:neha kumari**

**Roll no:2020331**

**Semester:IInd**