1. **Program to remove an object from the Stack when the position is passed as parameter.**

**Aim:**

To write a Program to remove an object from the Stack when the position is passed as parameter.

**Source code:**

**stack.java**

import java.io.\*;

import java.util.\*;

public class stack

{

public static void main(String []args)

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in)); Stack <String> s= new Stack<String>();

try

{

System.out.println("enter 5 string values");

for(int i=0;i<5;i++)

s.add(br.readLine());

System.out.println("Stack elements are:"+s);

int pos=0;

System.out.println("enter the position of element to remove object:");

pos=Integer.parseInt(br.readLine());

String st= s.remove(pos);

System.out.println("Stack element removed:"+st);

System.out.println("Remaining Stack elements are:"+s);

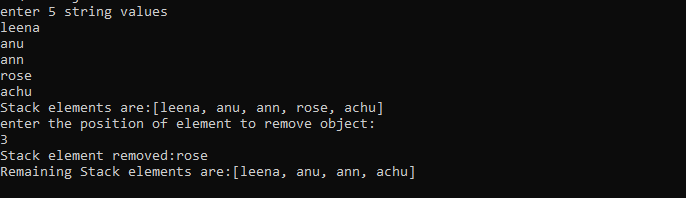
}

catch(IOException e)

{System.out.println(e);}

}}

**OUTPUT**



1. **Program to demonstrate the creation of queue object using the PriorityQueue class.**

**Aim:**

To write a program to demonstrate the creation of queue object using the PriorityQueue class.

**Source code:**

**PQueue.java**

import java.util.\*;

class PQueue {

public static void main(String args[])

{

PriorityQueue<Integer> pQueue = new PriorityQueue<Integer>();

// Adding items to the pQueue using add()

pQueue.add(10);

pQueue.add(20);

pQueue.add(15);

pQueue.add(50);

System.out.println("Queue elements:"+pQueue);

System.out.println("First element:"+pQueue.peek());

System.out.println("Element removed:"+pQueue.poll());

System.out.println("Remaining Queue:"+pQueue); //printing queue”);

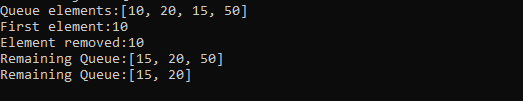
pQueue.remove(50);

System.out.println("Remaining Queue:"+pQueue); //printing queue”);

}

}

**OUTPUT**



1. **Program to demonstrate the addition and deletion of elements in deque.**

**Aim:**

To write a program to demonstrate the addition and deletion of elements in deque.

**Source code:**

**deque.java**

import java.util.\*;

public class deque

{

public static void main(String[] args)

{

Deque<String> dq= new LinkedList<String>();

// Add at the last

dq.add("Element 1 (Tail)");

// Add at the first

dq.addFirst("Element 2 (Head)");

// Add at the last

dq.addLast("Element 3 (Tail)");

// Add at the first

dq.push("Element 4 (Head)");

// Add at the last

dq.offer("Element 5 (Tail)");

// Add at the first

dq.offerFirst("Element 6 (Head)");

System.out.println(dq + "\n");

// We can remove the first element

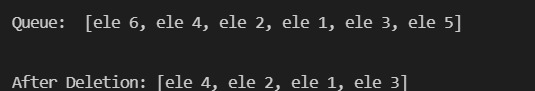
// or the last element.

dq.removeFirst();

dq.removeLast();

System.out.println("Deque after removing " + "first and last: " + dq);} }

**OUTPUT**



1. **Program to demonstrate the creation of Set object using the LinkedHashset class.**

**Aim:**

To write a program to demonstrate the creation of set object using the linkedhashset class.

**Source code:**

**LHSet.java**

import java.util.LinkedHashSet;

public class LHSet

{

// Main Method

public static void main(String[] args)

{

LinkedHashSet<String> ls = new LinkedHashSet<String>();

// Adding element to LinkedHashSet

ls.add("A");

ls.add("B");

ls.add("C");

ls.add("D");

// This will not add new element as A already exists

ls.add("A");

ls.add("E");

System.out.println("Size of LinkedHashSet = " + ls.size());

System.out.println("Original LinkedHashSet:" + ls);

System.out.println("Removing D from LinkedHashSet: " +ls.remove("D"));

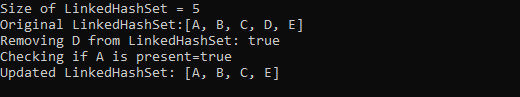
System.out.println("Checking if A is present=" + ls.contains("A"));

System.out.println("Updated LinkedHashSet: " + ls);

}

}

**OUTPUT**



1. **Write a Java program to compare two hash set.**

**Aim:**

To write a program to compare two hash set

**Source code:**

**hset.java**

import java.util.\*;

public class hset

{

public static void main(String[] args)

{

// Create a empty hash set

HashSet<String> h\_set = new HashSet<String>();

// use add() method to add values in the hash set

h\_set.add("Red");

h\_set.add("Green");

h\_set.add("Black");

h\_set.add("White");

h\_set.add("Yellow");

for (String element : h\_set)

System.out.println("1st set:"+element);

HashSet<String> h\_set2 = new HashSet<String>();

h\_set2.add("Red");

h\_set2.add("Pink");

h\_set2.add("Black");

h\_set2.add("Orange");

h\_set2.add("Yellow");

for (String element : h\_set2)

System.out.println("2st set:"+element);

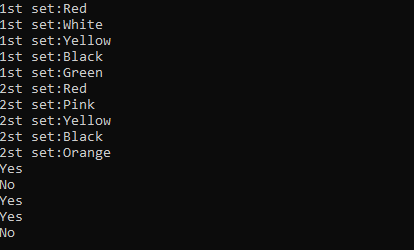
//comparison output in hash set

for (String element : h\_set)

System.out.println(h\_set2.contains(element) ? "Yes" : "No"); }

}

**OUTPUT**



1. **Program to demonstrate the working of Map interface by adding, changing and removing elements.**

**Aim:**

To write a program to demonstrate the working of Map interface by adding, changing and removing elements.

**Source code:**

**map.java**

import java.util.\*;

class map

{

public static void main(String args[])

{

HashMap<String, Integer> hm = new HashMap<String, Integer>();

hm.put("a",Integer(100));

hm.put("b", Integer(200));

hm.put("c", Integer(300));

hm.put("d", Integer(400));

// Traversing through the map, the insertion order is not retained in the hashmap.

System.out.print("initial map contents:");

for (HashMap.Entry<String, Integer> me : hm.entrySet())

{

System.out.print(me.getKey() + ":");

System.out.println(me.getValue());

}

System.out.print("map contents after updation:");

hm.put("c",new Integer(500));

for (HashMap.Entry<String, Integer> me : hm.entrySet())

{

System.out.print(me.getKey() + ":");

System.out.println(me.getValue());

}

hm.remove("a");

System.out.print("map contents after removal:");

for (HashMap.Entry<String, Integer> me : hm.entrySet())

{

System.out.print(me.getKey() + ":");

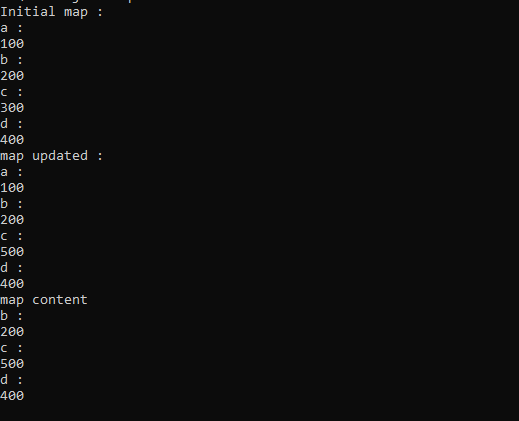
System.out.println(me.getValue());

}

}

}

**OUTPUT**



1. **Program to Convert HashMap to TreeMap.**

**Aim:**

To write a program to convert hashmap to treemap.

**Source code:**

**hashtree.java**

import java.util.HashMap;

import java.util.TreeMap;

import java.util.Map;

public class hashtree

{

public static void main(String[] a)

{

Map<String, String> map = new HashMap<String, String>();

map.put("1", "A");

map.put("2", "B");

map.put("4", "D");

map.put("3", "C");

map.put("5", "E");

map.put("6", "F");

map.put("8", "H");

map.put("7", "G");

map.put("9", "I");

System.out.println(" Map Elements = "+map);

Map<String, String>sorted = new TreeMap<String, String>(map); //shows sorted form only

System.out.println("Tree Map Elements = "+sorted);

}

}

**OUTPUT**



1. **Program to draw Circle, Rectangle, Line in Applet.**

**Aim:**

To write a program to draw Circle, Rectangle, Line in Applet.

**Source code:**

**first.java**

/\* applet program to draw line, circle, rectangle\*/

import java.applet.Applet;

import java.awt.Graphics;

public class first extends Applet

{

public void paint(Graphics g)

{

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

g.drawRect(100,60,40,30);

g.drawOval(130,10,50,50);

}

}

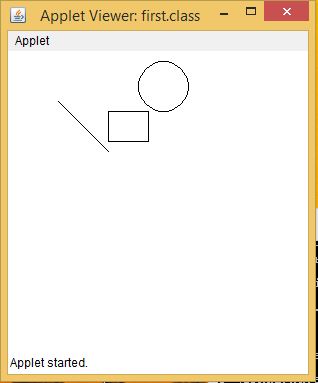
/\*

<applet code="first.class" width="300" height="300">

</applet>

\*/

**OUTPUT**



1. **Program to find maximum of three numbers using AWT.**

**Aim:**

To write a program to find maximum of three numbers using AWT

**Source code:**

**greatest.java**

import java.applet.\*;

import java.awt.\*;

import java.awt.event.\*;

public class greatest extends Applet implements ActionListener

{

TextField T1,T2,T3,T4;

Button B1;

public void init()

{

T1 = new TextField(10);

T2 = new TextField(10);

T3 = new TextField(10);

T4 = new TextField(10);

B1 = new Button("Click");

add(T1);

add(T2);

add(T3);

add(T4);

add(B1);

T1.setText("");

T2.setText("");

T3.setText("");

T4.setText("");

B1.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

int a,b,c,big=0;

a=Integer.parseInt(T1.getText());

b=Integer.parseInt(T2.getText());

c=Integer.parseInt(T3.getText());

if (a>b)

{

if (a>c)

big=a;

else

big=c;

}

else

{

if (b>c)

big=b;

else

big=c;

}

T4.setText(""+big);

}

}

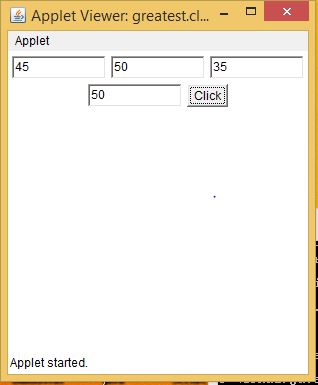
/\*

<applet code="greatest.class" width="300" height="300">

</applet>

\*/

**OUTPUT**



1. **Find the percentage of marks obtained by a student in 5 subjects. Display a happy face if he secures above 50% or a sad face if otherwise.**

**Aim:**

To write a program to find the percentage of marks obtained by a student in 5 subjects. Display a happy face if it is above 50% and sad face if not.

**Source code:**

**smiley.java**

import java.applet.\*;

import java.awt.\*;

import java.awt.event.\*;

public class smiley extends Applet implements ActionListener

{

TextField T1,T2,T3,T4,T5;

Label l,l2;

Button B1;

String str;

public smiley()

{

T1 = new TextField(10);

T2 = new TextField(10);

T3 = new TextField(10);

T4 = new TextField(10);

T5 = new TextField(10);

l = new Label("enter 5 numbers");

l2 = new Label();

B1 = new Button("Click");

add(l);

add(T1);

add(T2);

add(T3);

add(T4);

add(T5);

add(B1);

add(l2);

T1.setText("");

T2.setText("");

T3.setText("");

T4.setText("");

T5.setText("");

B1.addActionListener(this);

str="";

}

public void actionPerformed(ActionEvent e1)

{

int a,b,c,d,e,av=0;

a=Integer.parseInt(T1.getText());

b=Integer.parseInt(T2.getText());

c=Integer.parseInt(T3.getText());

d=Integer.parseInt(T2.getText());

e=Integer.parseInt(T3.getText());

av=(a+b+c+d+e)/5;

if(av>50)

str="yes";

else

str="no";

repaint();

}

public void paint(Graphics g)

{

super.paint(g);

g.drawString(str,550,550);

if(str=="yes")

{

g.drawOval(200,110,110,110);

g.setColor(Color.black);

g.fillOval(220,130,20,20);

g.fillOval(270,130,20,20);

g.drawArc(230,180,50,20,180,180);

}

else if(str=="no")

{

g.drawOval(200,110,110,110);

g.setColor(Color.black);

g.fillOval(220,130,20,20);

g.fillOval(270,130,20,20);

g.drawArc(230,180,50,20,-180,-180);

}

}

public static void main(String[] args)

{

new smiley();

}

}

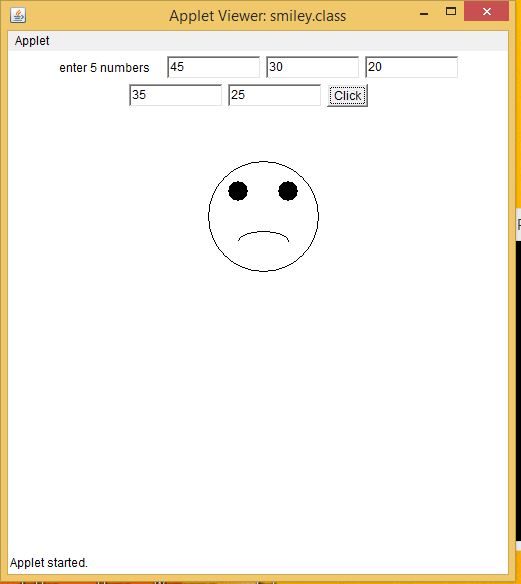
/\*

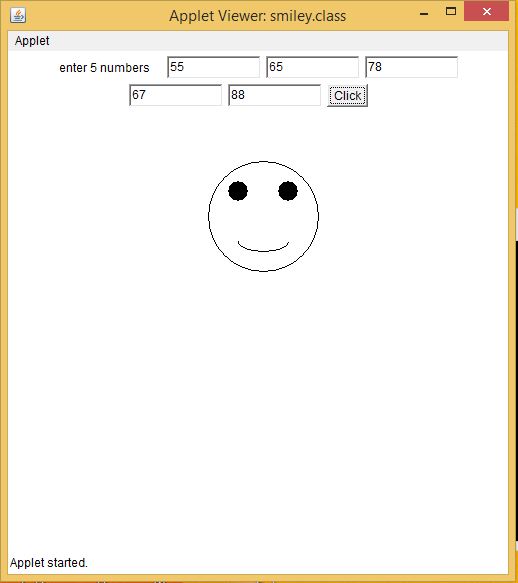
<applet code="smiley.class" width="500" height="500">

</applet>

\*/

**OUTPUT**





1. **Using 2D graphics commands in an Applet, construct a house. On mouse click event, change the color of the door from blue to red.**

**Aim:**

To write a program in 2D graphics command in an applet to construct a house. On mouse click event change the door colour.

**Source code:**

**home.java**

import java.applet.\*;

import java.awt.\*;

import java.awt.event.\*;

public class home extends Applet implements MouseListener

{

Color c =Color.blue;

public void init()

{

Label l;

l = new Label("hello");

l.setBounds(180,190,40,150);

add (l);

addMouseListener(this);

}

public void paint(Graphics g)

{

g.drawRect (100,150,200,200);

g.drawLine(100,150,200,50);

g.drawLine(300,150,200,50);

g.setColor(c);

g.fillRect (180,190,40,150);

}

public void mouseClicked(MouseEvent me)

{

c=Color.red;

// g.setColor(Color.red);

// g.fillRect (180,190,40,150);

repaint();

}

public void mouseExited(MouseEvent me){}

public void mouseMoved(MouseEvent me){}

public void mouseDragged(MouseEvent me){}

public void mouseEntered(MouseEvent me){}

public void mouseReleased(MouseEvent me){}

public void mousePressed(MouseEvent me){}

}

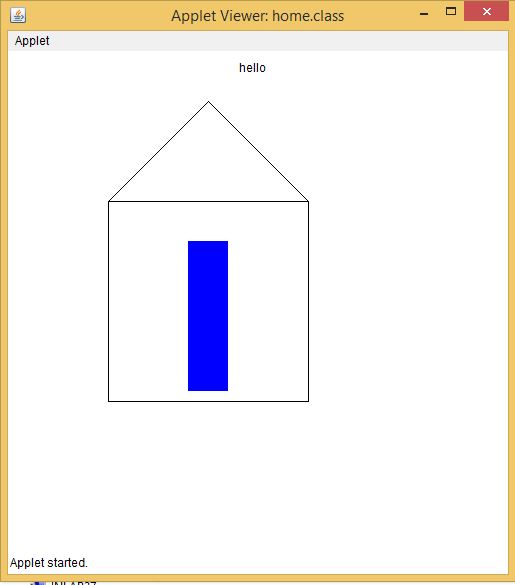
/\*

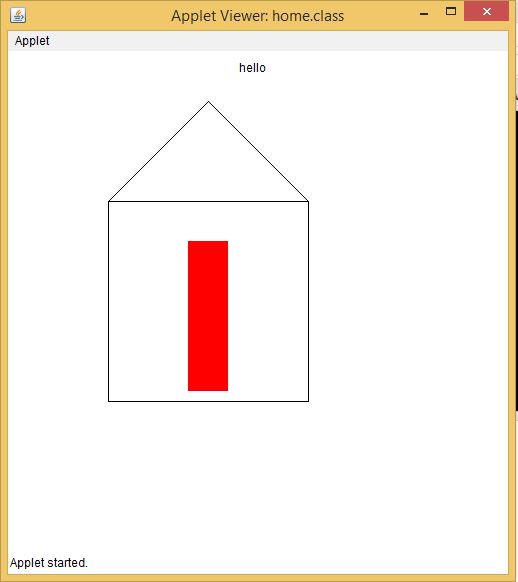
<applet code="home.class" width="500" height="500">

</applet>

\*/

**OUTPUT**





1. **Implement a simple calculator using AWT components.**

**Aim:**

To write a program to implement a simple calculator using AWT components

**Source code:**

**calc.java**

import java.awt.\*;

import java.awt.event.\*;

public class calc implements ActionListener

{

int c,n;

String s1,s2,s3,s4,s5;

Frame f;

Button b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14,b15,b16,b17;

Panel p;

TextField tf;

GridLayout g;

public calc()

{

f = new Frame("My calculator");

p = new Panel();

f.setLayout(new FlowLayout());

b1 = new Button("0");

b1.addActionListener(this);

b2 = new Button("1");

b2.addActionListener(this);

b3 = new Button("2");

b3.addActionListener(this);

b4 = new Button("3");

b4.addActionListener(this);

b5 = new Button("4");

b5.addActionListener(this);

b6 = new Button("5");

b6.addActionListener(this);

b7 = new Button("6");

b7.addActionListener(this);

b8 = new Button("7");

b8.addActionListener(this);

b9 = new Button("8");

b9.addActionListener(this);

b10 = new Button("9");

b10.addActionListener(this);

b11 = new Button("+");

b11.addActionListener(this);

b12 = new Button("-");

b12.addActionListener(this);

b13 = new Button("\*");

b13.addActionListener(this);

b14 = new Button("/");

b14.addActionListener(this);

b15 = new Button("%");

b15.addActionListener(this);

b16 = new Button("=");

b16.addActionListener(this);

b17 = new Button("C");

b17.addActionListener(this);

tf = new TextField(20);

f.add(tf);

g = new GridLayout(4,4,10,20);

p.setLayout(g);

p.add(b1);p.add(b2);p.add(b3);p.add(b4);p.add(b5);p.add(b6);p.add(b7);p.add(b8);p.add(b9); p.add(b10);p.add(b11);p.add(b12);p.add(b13);p.add(b14);p.add(b15);p.add(b16);p.add(b17);

f.add(p);

f.setSize(300,300);

f.setVisible(true);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==b1)

{

s3 = tf.getText();

s4 = "0";

s5 = s3+s4;

tf.setText(s5);

}

if(e.getSource()==b2)

{

s3 = tf.getText();

s4 = "1";

s5 = s3+s4;

tf.setText(s5);

}

if(e.getSource()==b3)

{

s3 = tf.getText();

s4 = "2";

s5 = s3+s4;

tf.setText(s5);

}if(e.getSource()==b4)

{

s3 = tf.getText();

s4 = "3";

s5 = s3+s4;

tf.setText(s5);

}

if(e.getSource()==b5)

{

s3 = tf.getText();

s4 = "4";

s5 = s3+s4;

tf.setText(s5);

}

if(e.getSource()==b6)

{

s3 = tf.getText();

s4 = "5";

s5 = s3+s4;

tf.setText(s5);

}

if(e.getSource()==b7)

{

s3 = tf.getText();

s4 = "6";

s5 = s3+s4;

tf.setText(s5);

}

if(e.getSource()==b8)

{

s3 = tf.getText();

s4 = "7";

s5 = s3+s4;

tf.setText(s5);

}

if(e.getSource()==b9)

{

s3 = tf.getText();

s4 = "8";

s5 = s3+s4;

tf.setText(s5);

}

if(e.getSource()==b10)

{

s3 = tf.getText();

s4 = "9";

s5 = s3+s4;

tf.setText(s5);

}

if(e.getSource()==b11)

{

s1 = tf.getText();

tf.setText("");

c=1;

}

if(e.getSource()==b12)

{

s1 = tf.getText();

tf.setText("");

c=2;

}

if(e.getSource()==b13)

{

s1 = tf.getText();

tf.setText("");

c=3;

}

if(e.getSource()==b14)

{

s1 = tf.getText();

tf.setText("");

c=4;

}

if(e.getSource()==b15)

{

s1 = tf.getText();

tf.setText("");

c=5;

}

if(e.getSource()==b16)

{

s2 = tf.getText();

if(c==1)

{

n = Integer.parseInt(s1)+Integer.parseInt(s2);

tf.setText(String.valueOf(n));

}

else

if(c==2)

{

n = Integer.parseInt(s1)-Integer.parseInt(s2);

tf.setText(String.valueOf(n));

}

else

if(c==3)

{

n = Integer.parseInt(s1)\*Integer.parseInt(s2);

tf.setText(String.valueOf(n));

}

if(c==4)

{

try

{

int p=Integer.parseInt(s2);

if(p!=0)

{

n = Integer.parseInt(s1)/Integer.parseInt(s2);

tf.setText(String.valueOf(n));

}

else

tf.setText("infinite");

}

catch(Exception i){}

}

if(c==5)

{

n = Integer.parseInt(s1)%Integer.parseInt(s2);

tf.setText(String.valueOf(n));

}

}

if(e.getSource()==b17)

{

tf.setText("");

}

}

public static void main(String[] abc)

{

calc v = new calc();

}

}

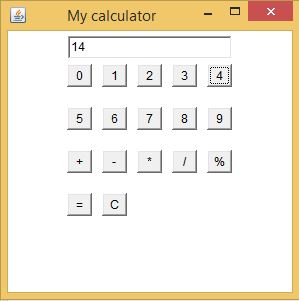
/\*

<applet code="calc.class" width="500" height="500">

</applet>

\*/

**OUTPUT**



1. **Develop a program that has a Choice component which contains the names of shapes such as rectangle, triangle, square and circle. Draw the corresponding shapes for given parameters as per user’s choice.**

**Aim:**

To develop a program that has choice components which contains names of the given shapes and draw the corresponding shapes as per users choice.

**Source code:**

**Drawchoice.java**

import java.awt.\*;

import java.awt.event.\*;

public class Drawchoice extends Frame implements ActionListener{

String data="";

String temp="";

Button b;

final Label label;

final Choice c;

public Drawchoice(){

b = new Button("Show");

b.addActionListener(this);

c = new Choice();

label = new Label();

c.setBounds(100, 100, 75, 75);

b.setBounds(200, 100, 50, 20);

label.setSize(400, 100);

label.setAlignment(Label.CENTER);

c.add("Circle");

c.add("Rectangle");

c.add("Line");

add(label);

add(c);

add(b);

setSize(400, 400);

setLayout(null);

setVisible(true);

}

public void paint(Graphics g)

{

super.paint(g);

if(data.equals("Line")){

g.drawLine(300,300,50,50);

}

else if(data.equals("Circle")){

g.drawOval(300,300,50,50);

}

else{

g.drawRect(300,300,50,50);

}

}

public void actionPerformed(ActionEvent e) {

if(e.getSource()==b){

temp = "Shape Selected: "+ c.getItem(c.getSelectedIndex());

data=c.getItem(c.getSelectedIndex());

label.setText(temp);

}

repaint();

}

public static void main(String[] args) {

new Drawchoice();

}

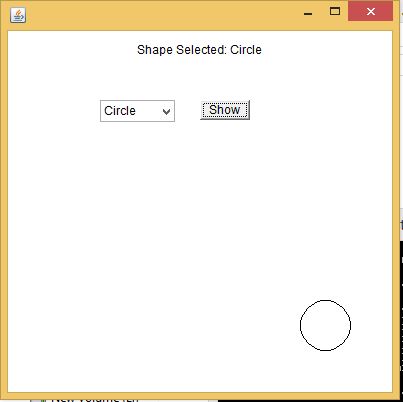
}

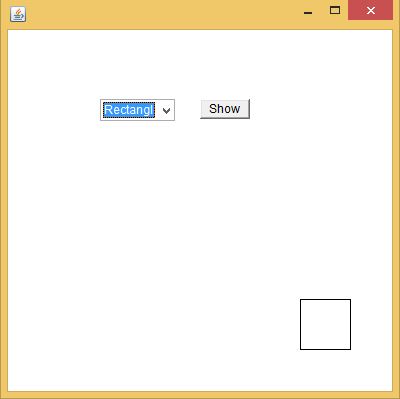
/\*

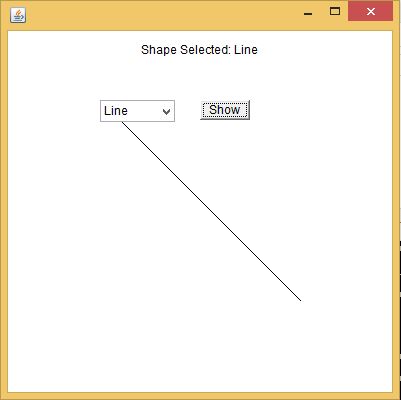
<applet code="Drawchoice.class" width="300" height="300">

</applet>\*/

**OUTPUT**







1. **Develop a program to handle all mouse events and window events.**

**Aim:**

To develop a program to handle all mouse events and window events.

**Source code:**

**mouseevents.java**

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

/\*<applet code="mouseevents" width=400 height=400></applet> \*/

public class mouseevents extends Applet implements MouseListener, MouseMotionListener

{

String msg = "";

int mouseX = 0, mouseY = 0; // coordinates of mouse

public void init()

{

addMouseListener(this);

addMouseMotionListener(this);

}

// Handle mouse clicked.

public void mouseClicked(MouseEvent me)

{

// save coordinates

mouseX = 0;

mouseY = 10;

msg = "Mouse clicked.";

repaint();

}

// Handle mouse entered.

public void mouseEntered(MouseEvent me)

{

// save coordinates

mouseX = 0;

mouseY = 10;

msg = "Mouse entered.";

repaint();

}

// Handle mouse exited.

public void mouseExited(MouseEvent me)

{

// save coordinates

mouseX = 0;

mouseY = 10;

msg = "Mouse exited.";

repaint();

}

// Handle button pressed.

public void mousePressed(MouseEvent me)

{

// save coordinates

mouseX = me.getX();

mouseY = me.getY();

msg = "Down";

repaint();

}

// Handle button released.

public void mouseReleased(MouseEvent me)

{

// save coordinates

mouseX = me.getX();

mouseY = me.getY();

msg = "Up";

repaint();

}

// Handle mouse dragged.

public void mouseDragged(MouseEvent me)

{

// save coordinates

mouseX = me.getX();

mouseY = me.getY();

msg = "\*";

showStatus("Dragging mouse at " + mouseX + ", " + mouseY);

repaint();

}

// Handle mouse moved.

public void mouseMoved(MouseEvent me)

{

// show status

showStatus("Moving mouse at " + me.getX() + ", " + me.getY());

}

// Display msg in applet window at current X,Y location.

public void paint(Graphics g)

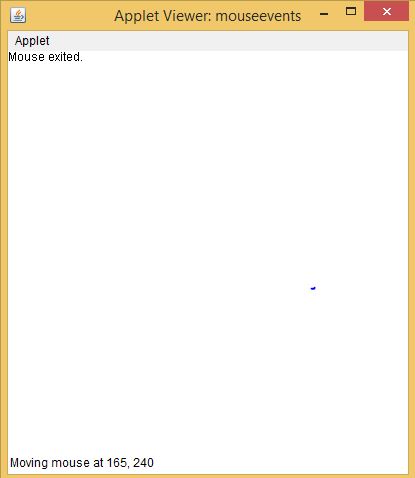
{

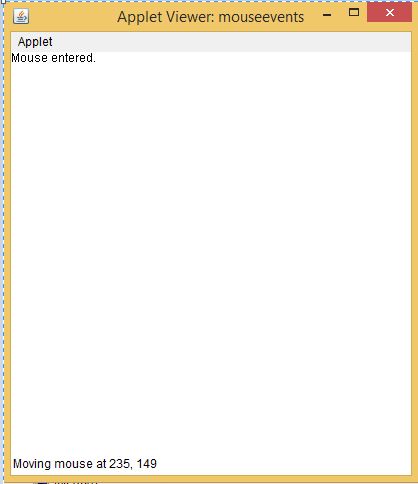
g.drawString(msg, mouseX, mouseY);

}

}

**OUTPUT**





**windowevents.java**

/\*Q windows events\*/

import java.awt.\*;

import java.awt.event.WindowEvent;

import java.awt.event.WindowListener;

public class windowevents extends Frame implements WindowListener

{

public windowevents()

{

addWindowListener(this);

setSize(400,400);

setLayout(null);

setVisible(true);

}

public static void main(String[] args)

{

new windowevents();

}

public void windowActivated(WindowEvent arg0)

{

System.out.println("activated");

}

public void windowClosed(WindowEvent arg0)

{

System.out.println("closed");

}

public void windowClosing(WindowEvent arg0)

{

System.out.println("closing");

dispose();

}

public void windowDeactivated(WindowEvent arg0)

{

System.out.println("deactivated");

}

public void windowDeiconified(WindowEvent arg0)

{

System.out.println("deiconified");

}

public void windowIconified(WindowEvent arg0)

{

System.out.println("iconified");

}

public void windowOpened(WindowEvent arg0)

{

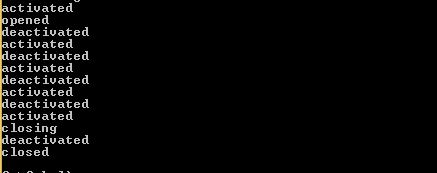
System.out.println("opened");

}

}

**OUTPUT**





1. **Develop a program to handle Key events.**

**Aim:**

To develop a program to handle key events

**Source code:**

**keyevents.java**

import java.applet.Applet;

import java.awt.\*;

import java.awt.event.\*;

/\* <APPLET CODE ="keyevents.class" WIDTH=500 HEIGHT=500></APPLET> \*/

public class keyevents extends Applet implements KeyListener

{

TextArea tpress,trel;

TextField t;

public void init()

{

t=new TextField(20);

t.addKeyListener(this);

tpress=new TextArea(3,70);

tpress.setEditable(false);

trel=new TextArea(3,70);

trel.setEditable(false);

add(t);

add(tpress);

add(trel);

}

public void keyTyped(KeyEvent e)

{

disppress(e,"Key Typed:");

}

public void keyPressed(KeyEvent e)

{

disppress(e,"KeyPressed:");

}

public void keyReleased(KeyEvent e)

{

String charString,keyCodeString;

char c=e.getKeyChar();

int keyCode=e.getKeyCode();

charString="Key character='"+c+"'"; keyCodeString="keycode="+keyCode+"("+KeyEvent.getKeyText(keyCode)+")";

trel.setText("Key released"+charString+keyCodeString);

}

protected void disppress(KeyEvent e,String s)

{

String charString,keyCodeString,tmpString;

char c=e.getKeyChar();

int keyCode=e.getKeyCode();

if(Character.isISOControl(c))

{

charString="key character=(an unprintable control character)";

}

else

{

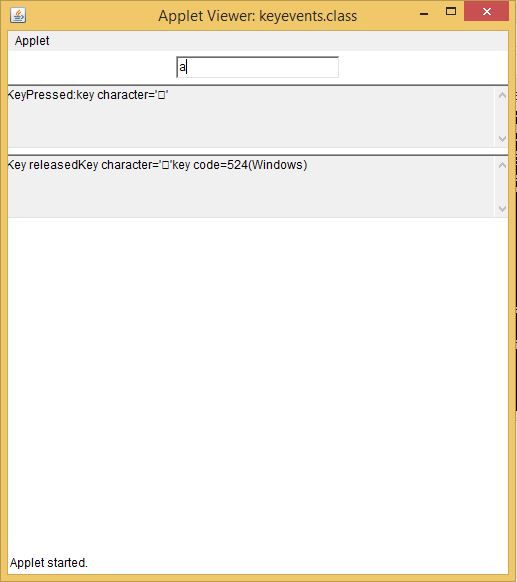
charString="key character='"+c+"'";

}

keyCodeString="keycode="+keyCode+"("+KeyEvent.getKeyText(keyCode)+")";

tpress.setText(s+charString);} }

**OUTPUT**



1. **Program to list the sub directories and files in a given directory and also search for a file name.**

**Aim:**

To write a program to list the sub directories and files in a given directory and also search for a file name.

**Source code:**

**dirfile.java**

import java.io.\*;

import java.lang.\*;

import java.io.File;

class dirfile

{

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

{

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

// creates a file object

File file = new File("C:\\java");

// returns an array of all files

String[] fileList = file.list();

for(String str : fileList)

{

System.out.println(str);

}

System.out.println("\n enter a file to search:");

String f = br.readLine();

for(String str : fileList)

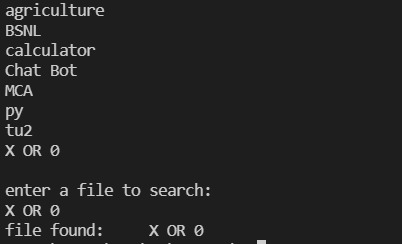
{

if(f.equals(str))

System.out.println("File found:"+str);

}}}

**OUTPUT**

****

1. **Write a program to write to a file, then read from the file and display the contents on the console.**

**Aim:**

To develop a program to write a file and then read from the file and display the contents on the console.

**Source code:**

**file1.java**

//writing and reading a file

import java.io.\*;

class file1

{

public static void main(String []a)

{

byte city[]={'D','E','L','H','I','\n','M','U','M','B','A','I','\n'};

FileOutputStream outfile=null; //writing to file

try

{

outfile= new FileOutputStream("city.txt");

outfile.write(city);

outfile.close();

}

catch(IOException e)

{

System.out.println(e);

}

FileInputStream infile=null; //reading from file

int b;

try

{

infile= new FileInputStream("city.txt");

while((b=infile.read())!= -1)

{

System.out.println((char)b);

}

infile.close();

}

catch(IOException e1)

{

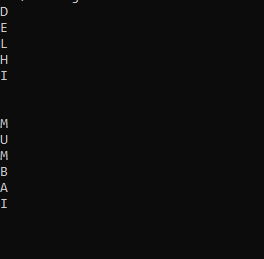
System.out.println(e1);

}

}

}

**OUTPUT**



1. **Write a program to copy one file to another.**

**Aim:**

To write a program to copy one file to another.

**Source code:**

**file2.java**

import java.io.\*;

class file2

{

public static void main(String []a)

{

FileInputStream infile=null;

FileOutputStream outfile=null;

byte b;

try

{

infile = new FileInputStream("city.txt");

outfile= new FileOutputStream("citynew.txt");

do

{

b=(byte) infile.read();

outfile.write(b);

System.out.println((char)b);

}while(b != -1);

}

catch(IOException e)

{

System.out.println(e);

//System.exit(-1);

}

finally

{

try

{

outfile.close();

infile.close();

}

catch(IOException e1)

{

System.out.println(e1);

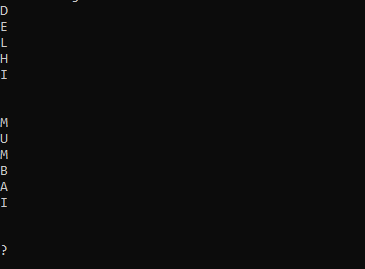
}

}

}

}

**OUTPUT**



1. **Write a program that reads from a file having integers. Copy even numbers and odd numbers to separate files.**

**Aim:**

To write a program that reads from a file having integers and copy even numbers and odd numbers and separate to another file.

**Source code:**

**file3.java**

import java.io.\*;

class file3

{

public static void main(String []a) throws IOException

{

int i;

File num= new File("integers.txt"); //creating main file

FileOutputStream fos=new FileOutputStream(num);

DataOutputStream dos= new DataOutputStream(fos); //for handling primitive data types

try

{

for(i=1;i<=10;i++)

dos.writeInt(i);

}

catch(IOException e)

{System.out.println("from 1"+e);}

dos.close();

fos.close();

FileInputStream fis=new FileInputStream(num);

DataInputStream dis= new DataInputStream(fis);

File num1= new File("odd.txt");

//creating odd number file

FileOutputStream fos1=new FileOutputStream(num1);

DataOutputStream dos1= new DataOutputStream(fos1);

File num2= new File("even.txt"); //creating even number file

FileOutputStream fos2=new FileOutputStream(num2);

DataOutputStream dos2= new DataOutputStream(fos2);

try

{

System.out.println("file content:");

for(int j=1;j<=10;j++)

{

i=dis.readInt();

System.out.println("inside fn:"+i);

if(i%2==0)

dos2.writeInt(i);

else

dos1.writeInt(i);

}

}

catch(IOException e1)

{System.out.println("from 2"+e1);}

dos1.close();

fos1.close();

dos2.close();

fos2.close();

dis.close();

fis.close();

FileInputStream fis1=new FileInputStream(num1);

DataInputStream dis1= new DataInputStream(fis1);

System.out.println("\nOdd file: ");

try

{

for(int j=1;j<=5;j++)

{

i=dis1.readInt();

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

}

}

catch(IOException e2)

{System.out.println("from 3"+e2);}

fis1.close();

dis1.close();

FileInputStream fis3=new FileInputStream(num2);

DataInputStream dis3= new DataInputStream(fis3);

System.out.println("\nEven file: ");

try

{

for(int k=1;k<=5;k++)

{

i=dis3.readInt();

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

}

}

catch(IOException e2)

{System.out.println("from 4"+e2);}

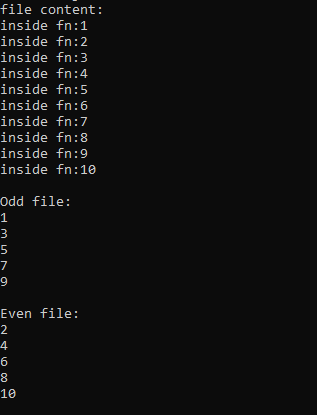
fis1.close();

dis1.close();

}

}

**OUTPUT**



1. **Client server communication using Socket – TCP/IP**

**Aim:**

To write a program for client server communication.

**Source code:**

**Tcpclient.java**

import java.io.\*;

import java.net.\*;

class Tcpclient

{

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

{

String s,ms;

BufferedReader infromuser = new BufferedReader(new InputStreamReader(System.in));

System.out.println("hai");

Socket clientsocket = new Socket("127.0.0.1",5000);

DataOutputStreamouttoserver=newDataOutputStream(clientsocket.getOuputStream());

System.out.println("\n enter a string:");

s=infromuser.readLine();

outtoserver.writeBytes(s+'\n');

BufferedReader infromserver = new BufferedReader(new InputStreamReader(clientsocket.getInputStream()));

ms=infromserver.readLine();

System.out.println("From server:"+ms);

clientsocket.close();

}

}

**Tcpserver.java**

import java.io.\*;

import java.net.\*;

class Tcpserver

{

public static void main(String []args) throws Exception

{

String s1,ms1;

ServerSocket serversocket = new ServerSocket(5000);

while(true)

{

Socket clientsocket = serversocket.accept();

BufferedReaderinfromclient=newBufferedReader(newInputStreamReder(clientsocket.getInputStream()));

s1=infromclient.readLine();

ms1=s1.toUpperCase()+'\n';

DataOutputStream outtoclient= new DataOutputStream(clientsocket.getOutputStream());

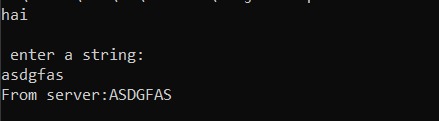
outtoclient.writeBytes(ms1);

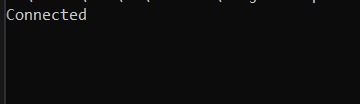
}

}

}

**OUTPUT**

****

****

1. **Client Server communication using DatagramSocket – UDP**

**Aim:**

To write a program for client server communication using datagram socket.

**Source code:**

**udpclient.java**

import java.io.\*;

import java.net.\*;

class udpclient

{

public static void main(String a[]) throws IOException

{

BufferedReader infromuser = new BufferedReader (new InputStreamReader(System.in));

DatagramSocket clientsocket = new DatagramSocket();

InetAddress ipaddress = InetAddress.getByName("127.0.0.1");

byte[] receivedata =new byte[1024];

byte[] senddata =new byte[1024];

System.out.println("\n enter a string:");

String str= infromuser.readLine();

senddata= str.getBytes();

DatagramPacket sendpacket = new DatagramPacket (senddata,senddata.length,ipaddress,5000);

clientsocket.send(sendpacket);

DatagramPacket receivepacket = new DatagramPacket (receivedata,receivedata.length);

clientsocket.receive(receivepacket);

String modified = new String(receivepacket.getData());

System.out.println("from Server"+modified);

clientsocket.close();

}

}

**udpserver.java**

import java.io.\*;

import java.net.\*;

class udpserver

{

public static void main(String a[]) throws IOException

{

DatagramSocket serversocket=new DatagramSocket(5000);

byte[] receivedata =new byte[1024];

byte[] senddata =new byte[1024];

while(true)

{

DatagramPacket receivepacket=new DatagramPacket(receivedata,receivedata.length);

serversocket.receive(receivepacket);

String sentence=new String(receivepacket.getData());

InetAddress ipaddress =receivepacket.getAddress();

int port=receivepacket.getPort();

String capital=sentence.toUpperCase();

System.out.println(capital);

senddata=capital.getBytes();

DatagramPacket sendpacket=new DatagramPacket(senddata,senddata.length,ipaddress,port);

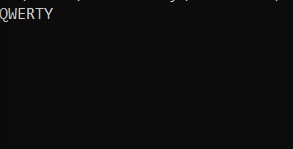
serversocket.send(sendpacket);

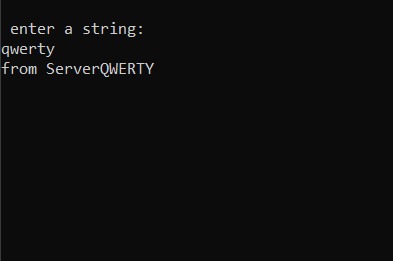
}

}

}

**OUTPUT**

****

****