**RANDOM NUMBERS**

**CODING**

**package** practical;

**import** java.util.Random;

**class** Prcs{

**privateint**num[]=**newint** [25];

**void** gen(){

Random r=**new** Random();

**for**(**int**i=0;i<25;i++){

num[i]=r.nextInt(25)+1;

}

}

**void** disp(){

System.***out***.println("The Generated Random Numbers are");

**for**(**int**i:num){

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

}

System.***out***.println("\n");

}

**void** check(){

**for**(**int**i=0;i<24;i++){

**if**(num[i]>num[i+1])

System.***out***.println("Number " + num[i] + " is Greater than " + num[i+1]);

**elseif**(num[i]<num[i+1])

System.***out***.println("Number " + num[i] + " is Lesser than " + num[i+1]);

**else**

System.***out***.println("Number " + num[i] + " is equal to " + num[i+1]);

}

}

}

**publicclass** RandomNumber {

**publicstaticvoid** main(String[] args) {

Prcs p=**new** Prcs();

p.gen();

p.disp();

p.check();

}

}

**OUTPUT**

The Generated Random Numbers are

11,13,7,22,11,3,11,7,20,5,23,22,15,6,8,15,9,22,4,22,17,3,3,2,19,

Number 11 is Lesser than 13

Number 13 is Greater than 7

Number 7 is Lesser than 22

Number 22 is Greater than 11

Number 11 is Greater than 3

Number 3 is Lesser than 11

Number 11 is Greater than 7

Number 7 is Lesser than 20

Number 20 is Greater than 5

Number 5 is Lesser than 23

Number 23 is Greater than 22

Number 22 is Greater than 15

Number 15 is Greater than 6

Number 6 is Lesser than 8

Number 8 is Lesser than 15

Number 15 is Greater than 9

Number 9 is Lesser than 22

Number 22 is Greater than 4

Number 4 is Lesser than 22

Number 22 is Greater than 17

Number 17 is Greater than 3

Number 3 is equal to 3

Number 3 is Greater than 2

Number 2 is Lesser than 19

**RATIONAL NUMBERS**

**CODING**

**package** practical;

**import** java.util.Scanner;

**class** Prcs{

**privateint**Num1,Num2,CmnFctNum;

**void** get(){

Scanner input=**new** Scanner(System.***in***);

System.***out***.print("Enter the First Number : ");

Num1=input.nextInt();

System.***out***.print("Enter the Second Number : ");

Num2=input.nextInt();

input.close();

}

**void** cmnFctr(){

CmnFctNum=gcd(Num1,Num2);

}

**privateint** gcd(**int**n1,**int**n2){

**if**(n2==0){

**return**n1;

}

**return** gcd(n2,n1%n2);

}

**void** disp(){

System.***out***.print("The Rational Number of " + Num1 + "/" + Num2 + " is " + (Num1/CmnFctNum) + "/" + (Num2/CmnFctNum));

}

}

**publicclass** RtnlNmbr {

**publicstaticvoid** main(String[] args) {

Prcs P1=**new** Prcs();

P1.get();

P1.cmnFctr();

P1.disp();

}

}

**OUTPUT**

Enter the First Number : 90

Enter the Second Number : 42

The Rational Number of 90/42 is 15/7

**EMPLOYEE INFORMATION USING PACKAGE**

**CODING**

**package** emplyInfo;

**import** java.util.Scanner;

**class** Prcs{

**private** String ID,Name;

**privateint**BP;

**privatefloat**DA,HRA,PF,NP,GP;

Prcs(){}

Prcs(String ID,String Name,**int**BP){

**this**.ID=ID;

**this**.Name=Name;

**this**.BP=BP;

}

Prcs(String ID,String Name){

**this**(ID,Name,5000);

}

Prcs(String ID,**int**BP){

**this**(ID,"xxx",BP);

}

Prcs(String ID){

**this**(ID,"xxx");

}

Prcs get(){

Prcs P=**new** Prcs();

String BPS;

Scanner input=**new** Scanner(System.***in***);

System.***out***.print("Enter the Employee ID : ");

ID=input.nextLine();

System.***out***.print("Enter the Employee Name : ");

Name=input.nextLine();

System.***out***.print("Enter the Basic Pay : ");

BPS=input.nextLine();

BP=Integer.*parseInt*(BPS.isEmpty()?"0":BPS);

input.close();

**if**(ID.isEmpty()){

System.***out***.println("Employee ID is Must");

System.*exit*(0);

}

**if**(Name.isEmpty()){

**if**(BP==0){

P=**new** Prcs(ID);

}

**else**{

P=**new** Prcs(ID,BP);

}

}

**elseif**(BP==0){

P=**new** Prcs(ID,Name);

}

**else**

P=**new** Prcs(ID,Name,BP);

**return**P;

}

**void** calc(){

DA=15\*BP/100;

HRA=12\*BP/100;

PF=8\*BP/100;

GP=BP+DA+HRA;

NP=GP-PF;

}

**void** disp(){

System.***out***.println("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.***out***.println("Employee Details");

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

System.***out***.println("Employee ID : " + ID);

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

System.***out***.println("Basic Pay : " + BP);

System.***out***.println("DA: " + DA);

System.***out***.println("HRA : " + HRA);

System.***out***.println("PF: " + PF);

System.***out***.println("Gross Pay : " + GP);

System.***out***.println("Net Pay : " + NP);

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

}

}

**publicclass** EmplyInfo {

**publicstaticvoid** main(String[] args) {

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

Prcs P1=**new** Prcs();

P1=P1.get();

P1.calc();

P1.disp();

}

}

**OUTPUT**

Enter the Employee ID : 1001

Enter the Employee Name : XXX

Enter the Basic Pay : 12000

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Employee Details

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Employee ID : 1001

Name: XXX

Basic Pay : 12000

DA: 1800.0

HRA : 1440.0

PF: 960.0

Gross Pay : 15240.0

Net Pay : 14280.0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**VEHICLE HIERARCHY**

**CODING**

**package** vehicleTest;

**class** Vehicle{

**int**tires,seats,doors;

}

**class** Car **extends** Vehicle{

Car(){

tires=4;

seats=5;

doors=4;

}

**public** String toString(){

**return** ("tires=" + tires + ", seats=" + seats + ", doors=" + doors);

}

}

**class** Bike **extends** Vehicle{

Bike(){

tires=2;

seats=2;

doors=0;

}

**public** String toString(){

**return** ("tires=" + tires + ", seats=" + seats + ", doors=" + doors);

}

}

**class** Bus **extends** Vehicle{

Bus(){

tires=6;

seats=64;

doors=3;

}

**public** String toString(){

**return** ("tires=" + tires + ", seats=" + seats + ", doors=" + doors);

}

}

**publicclass** VehicleTest {

**publicstaticvoid** main(String[] args) {

Vehicle v1=**new** Car(),v2=**new** Bike(),v3=**new** Bus();

System.***out***.println("Car have " + v1);

System.***out***.println("Bike have " + v2);

System.***out***.println("Bus have " + v3);

}

}

**OUTPUT**

Car have tires=4, seats=5, doors=4

Bike have tires=2, seats=2, doors=0

Bus have tires=6, seats=64, doors=3

**DATE CLASS**

**CODING**

**package** dateClass;

**class** Date{

**privateint**dt,mnt,yr,hr,min,sec;

Date(**int**yr,**int**mnt,**int**dt,**int**hr,**int**min,**int**sec){

Boolean bln=**new** Boolean(**true**);

**if**(yr<1 || mnt<1 || mnt>12 || hr<0 || hr>24 || min<0 || min>60 || sec<0 || sec>60 || dt<1){

bln=**false**;

}

**if**((((mnt%2==1 &&mnt<8) || mnt==8 || mnt==10 || mnt==12) &&dt>31) || (mnt==2 &&dt>29)

|| ((mnt==4 || mnt==6 || mnt==9 || mnt==11) &&dt>30) || (yr%4!=0 &&dt>28))

{

bln=**false**;

}

**if**(bln==**false**)

{

System.***err***.println("The given Date value is wrong");

System.*exit*(0);

}

**this**.dt=dt;

**this**.mnt=mnt;

**this**.yr=yr;

**this**.hr=hr;

**this**.min=min;

**this**.sec=sec;

}

Date(**int**yr,**int**mnt,**int**dt,**int**hr,**int**min){

**this**(yr,mnt,dt,hr,min,0);

}

Date(**int**yr,**int**mnt,**int**dt,**int**hr){

**this**(yr,mnt,dt,hr,0);

}

Date(**int**yr,**int**mnt,**int**dt){

**this**(yr,mnt,dt,0);

}

Date(**int**yr,**int**mnt){

**this**(yr,mnt,1);

}

Date(**int**yr){

**this**(yr,1);

}

Date(){

**this**(1990);

}

**void** setyr(**int**yr){

**this**.yr=yr;

}

**int** getyr(){

**return**yr;

}

**void** setmnt(**int**mnt){

**this**.mnt=mnt;

}

**int** getmnt(){

**return**mnt;

}

**void** setdt(**int**dt){

**this**.dt=dt;

}

**int** getdt(){

**return**dt;

}

**void** sethr(**int**hr){

**this**.hr=hr;

}

**int** gethr(){

**return**hr;

}

**void** setmin(**int**min){

**this**.min=min;

}

**int** getmin(){

**return**min;

}

**void** setsec(**int**sec){

**this**.sec=sec;

}

**int** getsec(){

**return**sec;

}

**public** String toString(){

String[] mntarr={"Jan","Feb","Mar","Abr","May","Jun","Jul","Aug","Sep",

"Oct","Nov","Dec"};

**return**dt+"-"+mntarr[mnt-1]+"-"+yr+","+hr+":"+min+":"+sec;

}

}

**publicclass** DateClass {

**publicstaticvoid** main(String[] args) {

Date dt1=**new** Date(2015,2,21,10);

System.***out***.println(dt1);

System.***out***.println("Year Seperately : " + dt1.getyr());

Date dt2=**new** Date();

System.***out***.println(dt2);

dt2.setmnt(2);

System.***out***.println(dt2);

}

}

**OUTPUT**

21-Feb-2015,10:0:0

Year Seperately : 2015

1-Jan-1990,0:0:0

1-Feb-1990,0:0:0

**STACK USING ARRAY AND LINKED LIST**

**CODING(ARRAY)**

import java.io.\*;

import java.util.\*;

interface stackintf

{

int n=50;

public void pop();

public void push();

public void display();

}

class Stack implements stackintf

{

int arr[]=new int[n];

int top=-1;

Scanner in=new Scanner(System.in);

public void push()

{

try

{

System.out.println("Enter The Element of Stack");

int elt=in.nextInt();

arr[++top]=elt;

}

catch (Exception e)

{

System.out.println("e");

}

}

public void pop()

{

int pop=arr[top];

top--;

System.out.println("Popped Element Is:"+pop);

}

public void display()

{

if(top<0)

{

System.out.println("Stack Is Empty");

}

else

{

System.out.print("Stack Elements Are:");

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

{

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

}

}

}

}

class StackADTArray

{

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

{

System.out.println("\nIMPLEMENTATION OF STACK USING ARRAY");

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

Scanner in=new Scanner(System.in);

Stack st=new Stack();

int no=0;

do

{

System.out.println("\n1.push \n2.pop \n3.display \n4.Exit\n");

System.out.print("Enter Your Choice:");

no=in.nextInt();

switch(no)

{

case 1:

st.push();

break;

case 2:

st.pop();

break;

case 3:

st.display();

break;

case 4:

System.exit(0);

}

}

while(no<=5);

}

}

**CODING (LINKED LIST)**

import java.io.\*;

import java.util.\*;

interface stackintf

{

public void pop();

public void push();

public void display();

}

class Stack implements stackintf

{

LinkedList list=new LinkedList();

Scanner in=new Scanner(System.in);

public void push()

{

try

{

System.out.print("Enter the Element of Stack:");

int elt=in.nextInt();

list.addLast(elt);

}

catch(Exception e)

{

System.out.println(e);

}

}

public void pop()

{

Object last=list.getLast();

list.removeLast();

System.out.println("Popped Element Is:"+last);

}

public void display()

{

if(list.isEmpty())

{

System.out.println("Stack Is Empty");

}

else

{

System.out.println("Stack Elements Are:"+list);

}

}

}

class StackADTLinkedList

{

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

{

System.out.println("\nIMPLEMENTATION OF STACK USING LINKED LIST");

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

Scanner in=new Scanner(System.in);

int no;

Stack st=new Stack();

do

{

System.out.println("\n1.push \n2.pop \n3.display \n4.Exit\n");

System.out.print("Enter Your Choice:");

no=in.nextInt();

switch(no)

{

case 1:

st.push();

break;

case 2:

st.pop();

break;

case 3:

st.display();

break;

case 4:

System.exit(0);

default:

System.out.println("Invalid Choice!");

}

}

while(no<=4);

}

}

**OUTPUT(ARRAY)**

\>java StackADTArray

IMPLEMENTATION OF STACK USING ARRAY

===================================

1.push

2.pop

3.display

4.Exit

Enter Your Choice:1

Enter the Element of Stack:4

1.push

2.pop

3.display

4.Exit

Enter Your Choice:1

Enter the Element of Stack:3

1.push

2.pop

3.display

4.Exit

Enter Your Choice:3

Stack Elements Are:[4, 3]

1.push

2.pop

3.display

4.Exit

Enter Your Choice:2

Popped Element Is:3

1.push

2.pop

3.display

4.Exit

Enter Your Choice:3

Stack Elements Are:[4]

1.push

2.pop

3.display

4.Exit

Enter Your Choice:4

**OUTPUT (LINKED LIST)**

\>java StackADTLinkedList

IMPLEMENTATION OF STACK USING LINKED LIST

=========================================

1.push

2.pop

3.display

4.Exit

Enter Your Choice:1

Enter the Element of Stack:5

1.push

2.pop

3.display

4.Exit

Enter Your Choice:1

Enter the Element of Stack:8

1.push

2.pop

3.display

4.Exit

Enter Your Choice:3

Stack Elements Are:[5, 8]

1.push

2.pop

3.display

4.Exit

Enter Your Choice:2

Popped Element Is:8

1.push

2.pop

3.display

4.Exit

Enter Your Choice:3

Stack Elements Are:[5]

1.push

2.pop

3.display

4.Exit

Enter Your Choice:4

**DRAWING SHAPES USING MENU**

**CODING**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package frmshp;

import java.awt.\*;

import javax.swing.\*;

public class FrmShp extends javax.swing.JFrame {

static FrmShp FS=new FrmShp();

/\*\*

Declaring the variables

\*/

String shp,clr;

/\*\*

\* Creates new form FrmShp

\*/

public FrmShp(){

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

mnuBr\_Shps = new javax.swing.JMenuBar();

mnu\_Shp = new javax.swing.JMenu();

mnu\_ShpRect = new javax.swing.JMenuItem();

mnu\_ShpOvl = new javax.swing.JMenuItem();

mnu\_ShpArc = new javax.swing.JMenuItem();

mnu\_FldShp = new javax.swing.JMenu();

mnu\_FldShpRect = new javax.swing.JMenuItem();

mnu\_FldShpOvl = new javax.swing.JMenuItem();

mnu\_FldShpArc = new javax.swing.JMenuItem();

mnu\_Clr = new javax.swing.JMenu();

mnu\_ClrRd = new javax.swing.JMenuItem();

mnu\_ClrGrn = new javax.swing.JMenuItem();

mnu\_ClrBlu = new javax.swing.JMenuItem();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

mnu\_Shp.setText("Shapes");

mnu\_ShpRect.setText("Rectangle");

mnu\_ShpRect.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

sbmnu\_ShpActionPerformed(evt);

}

});

mnu\_Shp.add(mnu\_ShpRect);

mnu\_ShpOvl.setText("Oval");

mnu\_ShpOvl.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

sbmnu\_ShpActionPerformed(evt);

}

});

mnu\_Shp.add(mnu\_ShpOvl);

mnu\_ShpArc.setText("Arc");

mnu\_ShpArc.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

sbmnu\_ShpActionPerformed(evt);

}

});

mnu\_Shp.add(mnu\_ShpArc);

mnuBr\_Shps.add(mnu\_Shp);

mnu\_FldShp.setText("Filled Shapes");

mnu\_FldShpRect.setText("Filled Rectangle");

mnu\_FldShpRect.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

sbmnu\_ShpActionPerformed(evt);

}

});

mnu\_FldShp.add(mnu\_FldShpRect);

mnu\_FldShpOvl.setText("Filled Oval");

mnu\_FldShpOvl.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

sbmnu\_ShpActionPerformed(evt);

}

});

mnu\_FldShp.add(mnu\_FldShpOvl);

mnu\_FldShpArc.setText("Filled Arc");

mnu\_FldShpArc.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

sbmnu\_ShpActionPerformed(evt);

}

});

mnu\_FldShp.add(mnu\_FldShpArc);

mnuBr\_Shps.add(mnu\_FldShp);

mnu\_Clr.setText("Colors");

mnu\_ClrRd.setText("Red");

mnu\_ClrRd.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

mnu\_ClrActionPerformed(evt);

}

});

mnu\_Clr.add(mnu\_ClrRd);

mnu\_ClrGrn.setText("Green");

mnu\_ClrGrn.setToolTipText("");

mnu\_ClrGrn.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

mnu\_ClrActionPerformed(evt);

}

});

mnu\_Clr.add(mnu\_ClrGrn);

mnu\_ClrBlu.setText("Blue");

mnu\_ClrBlu.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

mnu\_ClrActionPerformed(evt);

}

});

mnu\_Clr.add(mnu\_ClrBlu);

mnuBr\_Shps.add(mnu\_Clr);

setJMenuBar(mnuBr\_Shps);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 400, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 279, Short.MAX\_VALUE)

);

pack();

}// </editor-fold>

private void sbmnu\_ShpActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:\

Graphics g =getGraphics();

g.clearRect(0,50,getWidth(), getHeight());

if(clr=="red")

{

g.setColor(Color.red);

}

else if(clr=="green"){

g.setColor(Color.green);

}

else if(clr=="blue"){

g.setColor(Color.blue);

}

if(evt.getSource()==mnu\_ShpRect)

{

g.drawRect(100, 120, 100, 100);

}

else if(evt.getSource()==mnu\_ShpOvl){

g.drawOval(100,120,100,100);

}

else if(evt.getSource()==mnu\_ShpArc){

g.drawArc(100, 120, 100, 100, 90, 120);

}

else if(evt.getSource()==mnu\_FldShpRect){

g.fillRect(100,120,100,100);

}

else if(evt.getSource()==mnu\_FldShpOvl){

g.fillOval(100,120,100,100);

}

else if(evt.getSource()==mnu\_FldShpArc){

g.fillArc(100, 120, 100, 100, 90, 120);

}

}

private void mnu\_ClrActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

if(evt.getSource()==mnu\_ClrRd)

{

clr="red";

}

else if(evt.getSource()==mnu\_ClrGrn){

clr="green";

}

else if(evt.getSource()==mnu\_ClrBlu){

clr="blue";

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(FrmShp.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(FrmShp.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(FrmShp.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(FrmShp.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

SwingUtilities.invokeLater(() -> {

FS.setVisible(true);

FS.setBounds(Integer.parseInt(args[0]),Integer.parseInt(args[1]),Integer.parseInt(args[2]),Integer.parseInt(args[3]));

});

}

// Variables declaration - do not modify

private javax.swing.JMenuBar mnuBr\_Shps;

private javax.swing.JMenu mnu\_Clr;

private javax.swing.JMenuItem mnu\_ClrBlu;

private javax.swing.JMenuItem mnu\_ClrGrn;

private javax.swing.JMenuItem mnu\_ClrRd;

private javax.swing.JMenu mnu\_FldShp;

private javax.swing.JMenuItem mnu\_FldShpArc;

private javax.swing.JMenuItem mnu\_FldShpOvl;

private javax.swing.JMenuItem mnu\_FldShpRect;

private javax.swing.JMenu mnu\_Shp;

private javax.swing.JMenuItem mnu\_ShpArc;

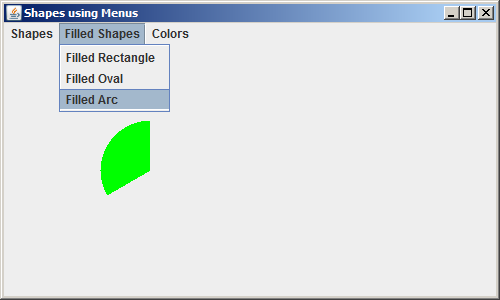
private javax.swing.JMenuItem mnu\_ShpOvl;

private javax.swing.JMenuItem mnu\_ShpRect;

// End of variables declaration

}

**OUTPUT**



**SCIENTIFIC CALCULATOR**

**CODING**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.lang.\*;

public class SimpleCalculator

{

public static void main(String[] args)

{

CalcFrame cf=new CalcFrame();

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

cf.setVisible(true);

}

}

class CalcFrame extends JFrame

{

public CalcFrame()

{

setTitle("CALCULATOR");

CalcPanel panel=new CalcPanel();

add(panel);

pack();

}

}

class CalcPanel extends JPanel

{

JButton display;

JPanel panel;

double result;

String lastcmd;

boolean start;

public CalcPanel()

{

setLayout(new BorderLayout());

result=0;

lastcmd="=";

start=true;

display=new JButton("0");

display.setEnabled(false);

add(display,BorderLayout.NORTH);

ActionListener insert=new InsertAction();

ActionListener cmd=new CommandAction();

panel=new JPanel();

panel.setLayout(new GridLayout(5,4));

addButton("1",insert);

addButton("2",insert);

addButton("3",insert);

addButton("/",cmd);

addButton("4",insert);

addButton("5",insert);

addButton("6",insert);

addButton("\*",cmd);

addButton("7",insert);

addButton("8",insert);

addButton("9",insert);

addButton("-",cmd);

addButton("0",insert);

addButton(".",insert);

addButton("pow",cmd);

addButton("+",cmd);

addButton("sin",insert);

addButton("cos",insert);

addButton("tan",insert);

addButton("=",cmd);

add(panel, BorderLayout.CENTER);

}

private void addButton(String label,ActionListener listener)

{

JButton button=new JButton(label);

button.addActionListener(listener);

panel.add(button);

}

private class InsertAction implements ActionListener

{

public void actionPerformed(ActionEvent ae)

{

String input=ae.getActionCommand();

if(start==true)

{

display.setText("");

start=false;

}

if(input.equals("sin"))

{

Double angle=Double.parseDouble(display.getText())\*2.0\*Math.PI/360.0;

display.setText(""+Math.sin(angle));

}

else if(input.equals("cos"))

{

Double angle=Double.parseDouble(display.getText())\*2.0\*Math.PI/360.0;

display.setText(""+Math.cos(angle));

}

else if(input.equals("tan"))

{

Double angle=Double.parseDouble(display.getText())\*2.0\*Math.PI/360.0;

display.setText(""+Math.tan(angle));

}

else

display.setText(display.getText()+input);

}

}

private class CommandAction implements ActionListener

{

public void actionPerformed(ActionEvent ae)

{

String command=ae.getActionCommand();

if(start==true)

{

if(command.equals("-"))

{

display.setText(command);

start=false;

}

else

lastcmd=command;

}

else

{

calc(Double.parseDouble(display.getText()));

lastcmd=command;

start=true;

}

}

}

public void calc(double x)

{

if(lastcmd.equals("+"))

result=result+x;

else if(lastcmd.equals("-"))

result=result-x;

else if(lastcmd.equals("\*"))

result=result\*x;

else if(lastcmd.equals("/"))

result=result/x;

else if(lastcmd.equals("="))

result=x;

else if(lastcmd.equals("pow"))

{

double powval=1.0;

for(double i=0.0;i<x;i++)

powval=powval\*result;

result=powval;

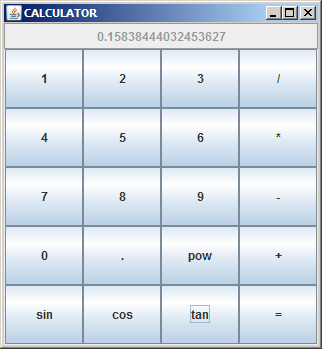
}

display.setText(""+ result);

}

}

**OUTPUT**



**MOUSE EVENTS**

**CODING**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package mouseevent;

public class MouseEvent extends javax.swing.JFrame {

/\*\*

\* Creates new form MouseEvent

\*/

public MouseEvent() {

super("Mouse Events");

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

sttsBr = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

addMouseMotionListener(new java.awt.event.MouseMotionAdapter() {

public void mouseDragged(java.awt.event.MouseEvent evt) {

mouseEvents(evt);

}

public void mouseMoved(java.awt.event.MouseEvent evt) {

mouseEvents(evt);

}

});

addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

mouseEvents(evt);

}

public void mouseEntered(java.awt.event.MouseEvent evt) {

mouseEvents(evt);

}

public void mouseExited(java.awt.event.MouseEvent evt) {

mouseEvents(evt);

}

public void mousePressed(java.awt.event.MouseEvent evt) {

mouseEvents(evt);

}

public void mouseReleased(java.awt.event.MouseEvent evt) {

mouseEvents(evt);

}

});

sttsBr.setText("Mouse Actions");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(sttsBr, javax.swing.GroupLayout.DEFAULT\_SIZE, 400, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addGap(0, 286, Short.MAX\_VALUE)

.addComponent(sttsBr))

);

pack();

}// </editor-fold>

private void mouseEvents(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

sttsBr.setText(evt.paramString());

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

}

catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(MouseEvent.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(MouseEvent.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(MouseEvent.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(MouseEvent.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new MouseEvent().setVisible(true);

}

});

}

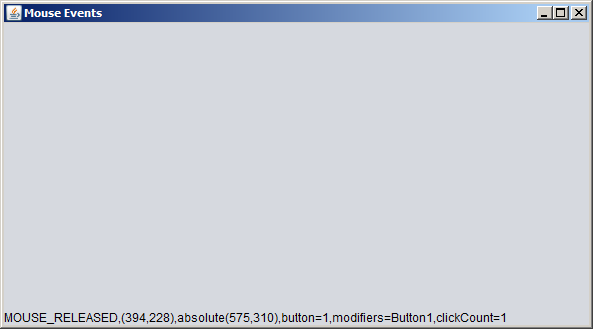
// Variables declaration - do not modify

private javax.swing.JLabel sttsBr;

// End of variables declaration

}

**OUTPUT**



**MONEY CONVERSION**

**CODING (CURRENCY)**

import java.io.\*;

public class Currency {

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

{

Dollar d= new Dollar();

Rupee r =new Rupee();

d.printDollar();

r.printRupee();

d.convertRupee();

// Write these contents into a text file using object serialization.

File f = new File("new.txt");

FileOutputStream fos = new FileOutputStream(f);

ObjectOutputStream oos = new ObjectOutputStream(fos);

oos.writeObject(d);

oos.writeObject(r);

oos.flush();

oos.close();

System.out.println(" Reading fform the file");

ObjectInputStream ois = new ObjectInputStream(new FileInputStream(f));

Dollar d1;

d1 = (Dollar)ois.readObject();

d1.printDollar();

Rupee r1;

r1 = (Rupee)ois.readObject();

r1.printRupee();

}

}

**CODING (DOLLAR)**

import java.io.\*;

class Dollar implements Serializable{

private String sym;

private double dol;

public Dollar() throws IOException{

InputStreamReader stream = new InputStreamReader(System.in);

BufferedReader bf= new BufferedReader(stream);

System.out.println(" Enter the Symbol for Dollar ");

sym= bf.readLine();

System.out.println (" Enter the amount in Dollar ");

dol= Double.parseDouble(bf.readLine());

}

void printDollar()

{

System.out.println(sym + dol + "\n");

}

void convertRupee()

{

dol = dol \* 61.58;

sym = "Rs.";

System.out.println(" The amount of Dollar in Rupees= " + sym + dol);

}

}

**CODING (RUPEES)**

import java.io.\*;

class Rupee implements Serializable{

private String sym;

private double rs;

public Rupee() throws IOException{

InputStreamReader stream = new InputStreamReader(System.in);

BufferedReader bf= new BufferedReader(stream);

System.out.println(" Enter the Symbol for Rupee ");

sym= bf.readLine();

System.out.println (" Enter the amount in Rupee ");

rs= Double.parseDouble(bf.readLine());

}

void printRupee()

{

System.out.println(sym + rs + "\n");

}

}

**OUTPUT**

\>java Currency

Enter the Symbol for Dollar

$

Enter the amount in Dollar

67

Enter the Symbol for Rupee

Rs.

Enter the amount in Rupee

45

$67.0

Rs.45.0

The amount of Dollar in Rupees= Rs.4125.86

Reading fform the file

Rs.4125.86

Rs.45.0

**SI UNITS**

**CODING**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package \_siunits;

public class SIUnits extends javax.swing.JFrame {

/\*\*

\* Creates new form SIUnits

\*/

public SIUnits() {

super("SI Units");

initComponents();

cbo\_UnitActionPerformed(null);

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jLabel1 = new javax.swing.JLabel();

cbo\_Unit = new javax.swing.JComboBox();

jLabel2 = new javax.swing.JLabel();

cbo\_Frm = new javax.swing.JComboBox();

jLabel3 = new javax.swing.JLabel();

cbo\_To = new javax.swing.JComboBox();

jLabel4 = new javax.swing.JLabel();

txt\_Vlu = new javax.swing.JTextField();

btn\_OK = new javax.swing.JButton();

jLabel5 = new javax.swing.JLabel();

lbl\_Ans = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jLabel1.setText("Select Your SI Unit");

cbo\_Unit.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Length", "Weight", "Time" }));

cbo\_Unit.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

cbo\_UnitActionPerformed(evt);

}

});

jLabel2.setText("From");

cbo\_Frm.setModel(new javax.swing.DefaultComboBoxModel(new String[] { " " }));

jLabel3.setText("To");

cbo\_To.setModel(new javax.swing.DefaultComboBoxModel(new String[] { " " }));

jLabel4.setText("Enter the Value");

btn\_OK.setText("OK");

btn\_OK.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btn\_OKActionPerformed(evt);

}

});

jLabel5.setText("Answer");

lbl\_Ans.setText(" ");

lbl\_Ans.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel1)

.addGroup(layout.createSequentialGroup()

.addGap(53, 53, 53)

.addComponent(jLabel2)))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(cbo\_Unit, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel3)))

.addGroup(layout.createSequentialGroup()

.addGap(53, 53, 53)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel4)

.addComponent(jLabel5))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(txt\_Vlu, javax.swing.GroupLayout.PREFERRED\_SIZE, 104, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(btn\_OK))

.addComponent(lbl\_Ans, javax.swing.GroupLayout.PREFERRED\_SIZE, 154, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGroup(layout.createSequentialGroup()

.addGap(19, 19, 19)

.addComponent(cbo\_Frm, javax.swing.GroupLayout.PREFERRED\_SIZE, 107, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(cbo\_To, javax.swing.GroupLayout.PREFERRED\_SIZE, 105, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(92, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1)

.addComponent(cbo\_Unit, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(29, 29, 29)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel2)

.addComponent(jLabel3))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(cbo\_Frm, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(cbo\_To, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel4)

.addComponent(txt\_Vlu, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(btn\_OK))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel5)

.addComponent(lbl\_Ans))

.addContainerGap(121, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

private void cbo\_UnitActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

if(cbo\_Unit.getSelectedItem()=="Length"){

cbo\_Frm.removeAllItems();

cbo\_Frm.addItem("Meter");

cbo\_Frm.addItem("Feet");

cbo\_Frm.addItem("Inch");

cbo\_To.removeAllItems();

cbo\_To.addItem("Meter");

cbo\_To.addItem("Feet");

cbo\_To.addItem("Inch");

}

else if(cbo\_Unit.getSelectedItem()=="Weight"){

cbo\_Frm.removeAllItems();

cbo\_Frm.addItem("Milli Gram");

cbo\_Frm.addItem("Gram");

cbo\_Frm.addItem("Kilo Gram");

cbo\_To.removeAllItems();

cbo\_To.addItem("Milli Gram");

cbo\_To.addItem("Gram");

cbo\_To.addItem("Kilo Gram");

}

else if(cbo\_Unit.getSelectedItem()=="Time"){

cbo\_Frm.removeAllItems();

cbo\_Frm.addItem("Second");

cbo\_Frm.addItem("Minute");

cbo\_Frm.addItem("Hour");

cbo\_To.removeAllItems();

cbo\_To.addItem("Second");

cbo\_To.addItem("Minute");

cbo\_To.addItem("Hour");

}

}

private void btn\_OKActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

if(cbo\_Frm.getSelectedItem()==cbo\_To.getSelectedItem()){

lbl\_Ans.setText(txt\_Vlu.getText());

}

else if(cbo\_Frm.getSelectedItem()=="Meter"){

if(cbo\_To.getSelectedItem()=="Feet"){

lbl\_Ans.setText(Double.toString(Float.parseFloat(txt\_Vlu.getText())\*3.28084));

}

if(cbo\_To.getSelectedItem()=="Inch"){

lbl\_Ans.setText(Double.toString(Float.parseFloat(txt\_Vlu.getText())\*39.3701));

}

}

else if(cbo\_Frm.getSelectedItem()=="Feet"){

if(cbo\_To.getSelectedItem()=="Meter"){

lbl\_Ans.setText(Double.toString(Float.parseFloat(txt\_Vlu.getText())\*0.3048));

}

if(cbo\_To.getSelectedItem()=="Inch"){

lbl\_Ans.setText(Double.toString(Float.parseFloat(txt\_Vlu.getText())\*12));

}

}

else if(cbo\_Frm.getSelectedItem()=="Inch"){

if(cbo\_To.getSelectedItem()=="Feet"){

lbl\_Ans.setText(Double.toString(Float.parseFloat(txt\_Vlu.getText())\*(1/12)));

}

if(cbo\_To.getSelectedItem()=="Meter"){

lbl\_Ans.setText(Double.toString(Float.parseFloat(txt\_Vlu.getText())\*0.0254));

}

}

else if(cbo\_Frm.getSelectedItem()=="Milli Gram"){

if(cbo\_To.getSelectedItem()=="Gram"){

lbl\_Ans.setText(Double.toString(Float.parseFloat(txt\_Vlu.getText())/1000));

}

if(cbo\_To.getSelectedItem()=="Kilo Gram"){

lbl\_Ans.setText(Double.toString(Float.parseFloat(txt\_Vlu.getText())/1000000));

}

}

else if(cbo\_Frm.getSelectedItem()=="Gram"){

if(cbo\_To.getSelectedItem()=="Milli Gram"){

lbl\_Ans.setText(Double.toString(Float.parseFloat(txt\_Vlu.getText())\*1000));

}

if(cbo\_To.getSelectedItem()=="Kilo Gram"){

lbl\_Ans.setText(Double.toString(Float.parseFloat(txt\_Vlu.getText())/1000));

}

}

else if(cbo\_Frm.getSelectedItem()=="Kilo Gram"){

if(cbo\_To.getSelectedItem()=="Milli Gram"){

lbl\_Ans.setText(Double.toString(Float.parseFloat(txt\_Vlu.getText())\*1000000));

}

if(cbo\_To.getSelectedItem()=="Gram"){

lbl\_Ans.setText(Double.toString(Float.parseFloat(txt\_Vlu.getText())\*1000));

}

}

else if(cbo\_Frm.getSelectedItem()=="Second"){

if(cbo\_To.getSelectedItem()=="Minute"){

lbl\_Ans.setText(Double.toString(Float.parseFloat(txt\_Vlu.getText())/60));

}

if(cbo\_To.getSelectedItem()=="Hour"){

lbl\_Ans.setText(Double.toString(Float.parseFloat(txt\_Vlu.getText())/360));

}

}

else if(cbo\_Frm.getSelectedItem()=="Minute"){

if(cbo\_To.getSelectedItem()=="Second"){

lbl\_Ans.setText(Double.toString(Float.parseFloat(txt\_Vlu.getText())\*60));

}

if(cbo\_To.getSelectedItem()=="Hour"){

lbl\_Ans.setText(Double.toString(Float.parseFloat(txt\_Vlu.getText())/60));

}

}

else if(cbo\_Frm.getSelectedItem()=="Hour"){

if(cbo\_To.getSelectedItem()=="Second"){

lbl\_Ans.setText(Double.toString(Float.parseFloat(txt\_Vlu.getText())\*360));

}

if(cbo\_To.getSelectedItem()=="Minute"){

lbl\_Ans.setText(Double.toString(Float.parseFloat(txt\_Vlu.getText())\*60));

}

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(SIUnits.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(SIUnits.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(SIUnits.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(SIUnits.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new SIUnits().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton btn\_OK;

private javax.swing.JComboBox cbo\_Frm;

private javax.swing.JComboBox cbo\_To;

private javax.swing.JComboBox cbo\_Unit;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

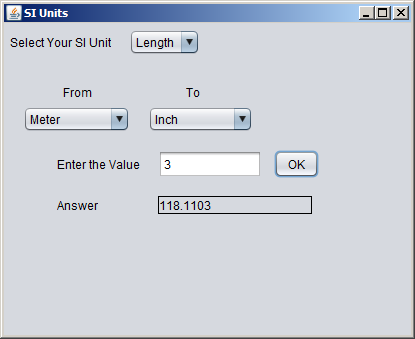
private javax.swing.JLabel lbl\_Ans;

private javax.swing.JTextField txt\_Vlu;

// End of variables declaration

}

**OUTPUT**



**INHERITANCE CONCEPTS WITH SHAPES**

**CODING(SHAPE)**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package shapes;

abstract public class Shapes {

protected String color;

protected boolean filled;

Shapes(){

color="Blue";

filled=false;

}

Shapes(String color,boolean filled){

this.color=color;

this.filled=filled;

}

String getColor(){

return color;

}

void setColor(String color){

this.color=color;

}

boolean isfilled(){

return filled;

}

void setFilled(boolean filled){

this.filled=filled;

}

public double getArea(){

return 0;

}

public double getPerimeter(){

return 0;

}

@Override

public String toString(){

return ("color=" + color + ", filled=" + filled);

}

}

**CODING(CIRCLE)**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package shapes;

public class Circle extends Shapes{

private double radius;

Circle(){

radius=4;

}

Circle(double radius){

this.radius=radius;

}

Circle(double radius,String color,boolean filled){

this.radius=radius;

this.color=color;

this.filled=filled;

}

double getRadius(){

return radius;

}

void setRadius(double radius){

this.radius=radius;

}

@Override

public double getArea(){

return (Math.PI \* radius \* radius);

}

@Override

public double getPerimeter(){

return (2 \* Math.PI \* radius);

}

@Override

public String toString(){

return ("radius=" + radius + ", color=" + color + ", filled=" + filled + ", Area=" + getArea() + ", Perimeter=" + getPerimeter());

}

}

**CODING (RECTANGLE)**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package shapes;

public class Rectangle extends Shapes{

protected double width;

protected double length;

Rectangle(){

width=2;

length=3;

}

Rectangle(double width,double length){

this.width=width;

this.length=length;

}

Rectangle(double width,double length,String color,boolean filled){

this.width=width;

this.length=length;

this.color=color;

this.filled=filled;

}

double getWidth(){

return width;

}

void setWidth(double width){

this.width=width;

}

double getLength(){

return length;

}

void setLength(double length){

this.length=length;

}

@Override

public double getArea(){

return (width \* length);

}

@Override

public double getPerimeter(){

return ((width + length) \* 2);

}

@Override

public String toString(){

return ("width=" + width + ", length=" + length + ", color=" + color + ", filled=" + filled + ", area=" + getArea() + ", perimeter=" + getPerimeter());

}

}

**CODING (SQUARE)**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package shapes;

public class Square extends Rectangle{

Square(){

width=length=4;

}

Square(double side){

width=length=side;

}

Square(double side,String color,boolean filled){

width=length=side;

this.color=color;

this.filled=filled;

}

double getSide(){

return width;

}

void setSide(double side){

width=length=side;

}

@Override

public void setWidth(double side){

setSide(side);

}

@Override

public void setLength(double side){

setSide(side);

}

@Override

public String toString(){

return ("width=" + width + ", length=" + length + ", color=" + color + ", filled=" + filled + ", area=" + getArea() + ", perimeter=" + getPerimeter());

}

}

**CODING (SHAPE TEST)**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package shapes;

public class ShapeTest {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

// TODO code application logic here

Circle c=new Circle();

Rectangle r=new Rectangle();

Square s=new Square();

System.out.println("Circle " + c);

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

System.out.println("Square " + s + "\n");

c=new Circle(3,"Red",true);

System.out.println("Circle " + c);

r=new Rectangle(4.0,4.5,"Green",true);

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

s=new Square(6.0,"Gray",true);

System.out.println("Square " + s);

}}

**OUTPUT**

Circle radius=4.0, color=Blue, filled=false, Area=50.26548245743669, Perimeter=25.132741228718345

Rectangle width=2.0, length=3.0, color=Blue, filled=false, area=6.0, perimeter=10.0

Square width=4.0, length=4.0, color=Blue, filled=false, area=16.0, perimeter=16.0

Circle radius=3.0, color=Red, filled=true, Area=28.274333882308138, Perimeter=18.84955592153876

Rectangle width=4.0, length=4.5, color=Green, filled=true, area=18.0, perimeter=17.0

Square width=6.0, length=6.0, color=Gray, filled=true, area=36.0, perimeter=24.0

**STUDENT INFORMATION USING TEXT FILE**

**CODING**

import java.io.\*;

class StudInfo

{

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

{

FileInputStream fis; FileOutputStream fos;

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

System.out.print("Enter the file name :");

String fname=br.readLine();

System.out.println("\*\*\*\*\*MENU\*\*\*\*\*");

System.out.println("1.Write\n2.Read\n3.Exit");

System.out.print("Enter your choice ?");

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

try

{

switch(ch)

{

case 1:

System.out.println("Writing Process Started...");

fos=new FileOutputStream(fname);

char c;

while((c=(char)br.read())!='#')

fos.write(c);

fos.close();

System.out.println("Writing process Completed.");

break;

case 2:

System.out.println("\nReading Process Started...\n");

fis=new FileInputStream(fname);

int ch1;

while((ch1=fis.read())!=-1)

System.out.print((char)ch1);

fis.close();

System.out.println("\n\nReading Process Completed.");

break;

case 3:

System.out.println("Process Terminated.");

System.exit(0);

break;

default :

System.out.println("Invalid Selection.");

}//switch

}

catch(Exception exp)

{

System.out.println("File Not Found!");

}

}//main()

}//class

**OUTPUT**

\>javac StudInfo.java

\>java StudInfo

Enter the file name : new.txt

\*\*\*\*\*MENU\*\*\*\*\*

1.Write

2.Read

3.Exit

Enter your choice ?1

Writing Process Started...

Name: MCA

Roll No: IIM

Class: KGiSL

Writing process Completed.

**TREE VIEW**

**CODING**

import javax.swing.JFrame;

import javax.swing.JTree;

import javax.swing.SwingUtilities;

import javax.swing.tree.DefaultMutableTreeNode;

public class TreeExample extends JFrame

{

private JTree tree;

public TreeExample()

{

DefaultMutableTreeNode root = new DefaultMutableTreeNode("Root");

DefaultMutableTreeNode vegetableNode = new DefaultMutableTreeNode("Vegetables");

DefaultMutableTreeNode fruitNode = new DefaultMutableTreeNode("Fruits");

DefaultMutableTreeNode v1 = new DefaultMutableTreeNode("Carrot");

DefaultMutableTreeNode v2 = new DefaultMutableTreeNode("Beetroot");

DefaultMutableTreeNode v3 = new DefaultMutableTreeNode("Potato");

DefaultMutableTreeNode f1 = new DefaultMutableTreeNode("Mango");

DefaultMutableTreeNode f2 = new DefaultMutableTreeNode("Banana");

DefaultMutableTreeNode f3 = new DefaultMutableTreeNode("Jack Fruit");

vegetableNode.add(v1);

vegetableNode.add(v2);

vegetableNode.add(v3);

fruitNode.add(f1);

fruitNode.add(f2);

fruitNode.add(f3);

//add the child nodes to the root node

root.add(vegetableNode);

root.add(fruitNode);

//create the tree by passing in the root node

tree = new JTree(root);

add(tree);

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

this.setTitle("JTree Example");

this.pack();

this.setVisible(true);

}

public static void main(String[] args)

{

SwingUtilities.invokeLater(new Runnable() {

@Override

public void run() {

new TreeExample();

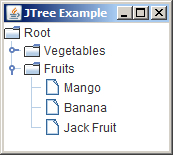
}

});

}

}

**OUTPUT**



**VIEW IMAGE**

**CODING**

import java.awt.\*;

import java.awt.image.\*;

import java.io.\*;

import javax.imageio.\*;

import javax.swing.\*;

public class ImageDemo

{

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

{

new ImageDemo(args[0]);

}

public ImageDemo(final String filename) throws Exception

{

SwingUtilities.invokeLater(new Runnable()

{

public void run()

{

JFrame editorFrame = new JFrame("Image Demo");

editorFrame.setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);

BufferedImage image = null;

try

{

image = ImageIO.read(new File(filename));

}

catch (Exception e)

{

e.printStackTrace();

System.exit(1);

}

ImageIcon imageIcon = new ImageIcon(image);

JLabel jLabel = new JLabel();

jLabel.setIcon(imageIcon);

editorFrame.getContentPane().add(jLabel, BorderLayout.CENTER);

editorFrame.pack();

editorFrame.setLocationRelativeTo(null);

editorFrame.setVisible(true);

}

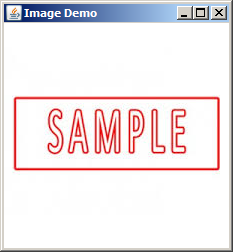
});

}

}

**OUTPUT**

\>java ImageDemo images.jpg



**ANIMATION USING MULTITHREADING**

**CODING**

import java.awt.Color;

import java.awt.Dimension;

import java.awt.FlowLayout;

import java.awt.Graphics;

import java.awt.Graphics2D;

import java.awt.RenderingHints;

import java.awt.image.BufferedImage;

import javax.swing.JComponent;

import javax.swing.JFrame;

import javax.swing.JPanel;

public class SlideShow extends JComponent {

private BufferedImage[] slide;

private Dimension slideSize;

private volatile int currSlide;

private Thread internalThread;

private volatile boolean noStopRequested;

public SlideShow() {

currSlide = 0;

slideSize = new Dimension(50, 50);

buildSlides();

setMinimumSize(slideSize);

setPreferredSize(slideSize);

setMaximumSize(slideSize);

setSize(slideSize);

noStopRequested = true;

Runnable r = new Runnable() {

public void run() {

try {

runWork();

} catch (Exception x) {

x.printStackTrace();

}

}

};

internalThread = new Thread(r, "SlideShow");

internalThread.start();}

private void buildSlides() {

RenderingHints renderHints = new RenderingHints(

RenderingHints.KEY\_ANTIALIASING,

RenderingHints.VALUE\_ANTIALIAS\_ON);

renderHints.put(RenderingHints.KEY\_RENDERING,

RenderingHints.VALUE\_RENDER\_QUALITY);

slide = new BufferedImage[20];

Color rectColor = Color.BLUE;

Color circleColor = Color.YELLOW;

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

slide[i] = new BufferedImage(slideSize.width, slideSize.height,

BufferedImage.TYPE\_INT\_RGB);

Graphics2D g2 = slide[i].createGraphics();

g2.setRenderingHints(renderHints);

g2.setColor(rectColor);

g2.fillRect(0, 0, slideSize.width, slideSize.height);

g2.setColor(circleColor);

int diameter = 0;

if (i < (slide.length / 2)) {

diameter = 5 + (8 \* i);

} else {

diameter = 5 + (8 \* (slide.length - i));

}

int inset = (slideSize.width - diameter) / 2;

g2.fillOval(inset, inset, diameter, diameter);

g2.setColor(Color.black);

g2.drawRect(0, 0, slideSize.width - 1, slideSize.height - 1);

g2.dispose();

}

}

public void paint(Graphics g) {

g.drawImage(slide[currSlide], 0, 0, this);

}

private void runWork() {

while (noStopRequested) {

try {

Thread.sleep(100); // 10 frames per second

currSlide = (currSlide + 1) % slide.length;

repaint();

} catch (InterruptedException x) {

Thread.currentThread().interrupt();

}

}

}

public void stopRequest() {

noStopRequested = false;

internalThread.interrupt();

}

public boolean isAlive() {

return internalThread.isAlive();

}

public static void main(String[] args) {

SlideShow ss = new SlideShow();

JPanel p = new JPanel(new FlowLayout());

p.add(ss);

JFrame f = new JFrame("SlideShow");

f.setContentPane(p);

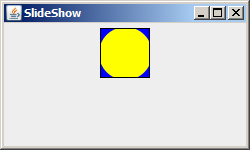
f.setSize(250, 150);

f.setVisible(true);

}

}

**OUTPUT**



**PRODUCER CONSUMER PROBLEM**

**CODING**

public class ProducerConsumerTest {

public static void main(String[] args) {

CubbyHole c = new CubbyHole();

Producer p1 = new Producer(c, 1);

Consumer c1 = new Consumer(c, 1);

p1.start();

c1.start();

}

}

class CubbyHole {

private int contents;

private boolean available = false;

public synchronized int get() {

while (available == false) {

try {

wait();

}

catch (InterruptedException e) {

}

}

available = false;

notifyAll();

return contents;

}

public synchronized void put(int value) {

while (available == true) {

try {

wait();

}

catch (InterruptedException e) {

}

}

contents = value;

available = true;

notifyAll();

}

}

class Consumer extends Thread {

private CubbyHole cubbyhole;

private int number;

public Consumer(CubbyHole c, int number) {

cubbyhole = c;

this.number = number;

}

public void run() {

int value = 0;

for (int i = 0; i < 10; i++) {

value = cubbyhole.get();

System.out.println("Consumer #"

+ this.number

+ " got: " + value);

}

}

}

class Producer extends Thread {

private CubbyHole cubbyhole;

private int number;

public Producer(CubbyHole c, int number) {

cubbyhole = c;

this.number = number;

}

public void run() {

for (int i = 0; i < 10; i++) {

cubbyhole.put(i);

System.out.println("Producer #" + this.number

+ " put: " + i);

try {

sleep((int)(Math.random() \* 100));

} catch (InterruptedException e) { }

}

}

}

**OUTPUT**

Producer #1 put: 0

Consumer #1 got: 0

Producer #1 put: 1

Consumer #1 got: 1

Producer #1 put: 2

Consumer #1 got: 2

Producer #1 put: 3

Consumer #1 got: 3

Producer #1 put: 4

Consumer #1 got: 4

Producer #1 put: 5

Consumer #1 got: 5

Producer #1 put: 6

Consumer #1 got: 6

Producer #1 put: 7

Consumer #1 got: 7

Producer #1 put: 8

Consumer #1 got: 8

Producer #1 put: 9

Consumer #1 got: 9

**MULTI-THREADING NUMBER GENERATION**

**CODING**

import java.util.\*;

import java.io.\*;

class Fibonacci extends Thread

{

private PipedWriter out=new PipedWriter();

public PipedWriter getPipedWriter()

{

return out;

}

public void run()

{

Thread t=Thread.currentThread();

t.setName("A1");

System.out.println(t.getName()+" Thread started...");

int fibo=0,fibo1=0,fibo2=1;

while(true)

{

try

{

fibo=fibo1+fibo2;

if(fibo>100000)

{

out.close();

break;

}

out.write(fibo);

sleep(1000);

}

catch(Exception e)

{

System.out.println("Exception:"+e);

}

fibo1=fibo2;

fibo2=fibo;

}

System.out.println(t.getName()+" Thread exiting...");

}

}

class Prime extends Thread

{

private PipedWriter out1=new PipedWriter();

public PipedWriter getPipedWriter()

{

return out1;

}

public void run()

{

Thread t=Thread.currentThread();

t.setName("Prime");

System.out.println(t.getName() +" Thread Started...");

int prime=1;

while(true)

{

try

{

if(prime>100000)

{

out1.close();

break;

}

if(isPrime(prime))

out1.write(prime);

prime++;

sleep(0);

}

catch(Exception e)

{

System.out.println(t.getName()+" Thread exiting...");

System.exit(0);

}

}

}

public boolean isPrime(int n)

{

int m=(int)Math.round(Math.sqrt(n));

if(n==1||n==2)

return true;

for(int i=2;i<=m;i++)

if(n%i==0)

return false;

return true;

}

}

public class MultiThreadDemo

{

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

{

Thread t=Thread.currentThread();

t.setName("Main");

System.out.println(t.getName()+" Thread Started...");

Fibonacci fibObj=new Fibonacci();

Prime primeObj=new Prime();

PipedReader pr=new PipedReader(fibObj.getPipedWriter());

PipedReader pr1=new PipedReader(primeObj.getPipedWriter());

fibObj.start();

primeObj.start();

int fib=pr.read(),prm=pr1.read();

System.out.println("The numbers common to PRIME and FIBONACCI:");

while((fib!=-1)&&(prm!=-1))

{

while(prm<=fib)

{

if(fib==prm)

System.out.println(prm);

prm=pr1.read();

}

fib=pr.read();

}

System.out.println(t.getName()+ " Thread exiting...");

}

}

**OUTPUT**

Main Thread Started...

A1 Thread started...

Prime Thread Started...

The numbers common to PRIME and FIBONACCI:

1

2

3

5

13

89

233

1597

28657

A1 Thread exiting...

Main Thread exiting...

Prime Thread exiting...

**TEXT MESSAGE USING SOCKET**

**CODING (SERVER)**

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.io.OutputStream;

import java.io.OutputStreamWriter;

import java.net.ServerSocket;

import java.net.Socket;

public class Server

{

private static Socket socket;

public static void main(String[] args)

{

try

{

int port = 25000;

ServerSocket serverSocket = new ServerSocket(port);

System.out.println("Server Started and listening to the port 25000");

//Server is running always. This is done using this while(true) loop

while(true)

{

//Reading the message from the client

socket = serverSocket.accept();

InputStream is = socket.getInputStream();

InputStreamReader isr = new InputStreamReader(is);

BufferedReader br = new BufferedReader(isr);

String number = br.readLine();

System.out.println("Message received from client is "+number);

//Multiplying the number by 2 and forming the return message

String returnMessage;

try

{

int numberInIntFormat = Integer.parseInt(number);

int returnValue = numberInIntFormat\*2;

returnMessage = String.valueOf(returnValue) + "\n";

}

catch(NumberFormatException e)

{

//Input was not a number. Sending proper message back to client.

returnMessage = "Please send a proper number\n";

}

//Sending the response back to the client.

OutputStream os = socket.getOutputStream();

OutputStreamWriter osw = new OutputStreamWriter(os);

BufferedWriter bw = new BufferedWriter(osw);

bw.write(returnMessage);

System.out.println("Message sent to the client is "+returnMessage);

bw.flush();

}

}

catch (Exception e)

{

e.printStackTrace();

}

finally

{

try

{

socket.close();

}

catch(Exception e){}

}

}

}

**Coding (Client):-**

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.io.OutputStream;

import java.io.OutputStreamWriter;

import java.net.InetAddress;

import java.net.Socket;

public class Client

{

private static Socket socket;

public static void main(String args[])

{

try

{

String host = "localhost";

int port = 25000;

InetAddress address = InetAddress.getByName(host);

socket = new Socket(address, port);

//Send the message to the server

OutputStream os = socket.getOutputStream();

OutputStreamWriter osw = new OutputStreamWriter(os);

BufferedWriter bw = new BufferedWriter(osw);

String number = "2";

String sendMessage = number + "\n";

bw.write(sendMessage);

bw.flush();

System.out.println("Message sent to the server : "+sendMessage);

//Get the return message from the server

InputStream is = socket.getInputStream();

InputStreamReader isr = new InputStreamReader(is);

BufferedReader br = new BufferedReader(isr);

String message = br.readLine();

System.out.println("Message received from the server : " +message);

}

catch (Exception exception)

{

exception.printStackTrace();

}

finally

{

//Closing the socket

try

{

socket.close();

}

catch(Exception e)

{

e.printStackTrace();

}

}

}

}

**OUTPUT (SERVER)**

\>java Server

Server Started and listening to the port 25000

Message received from client is 2

Message sent to the client is 4

**OUTPUT (CLIENT)**

\>java Client

Message sent to the server : 2

Message received from the server : 4

**FILE TRANSFER USING TCP**

**CODING (SERVER)**

import java.io.\*;

import java.net.\*;

class TCPServer{

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

{

String clientSentence;

String capitalizedSentence;

ServerSocket welcomeSocket = new ServerSocket(6789);

while(true)

{

Socket connectionSocket = welcomeSocket.accept();

BufferedReader inFromClient =

new BufferedReader(new InputStreamReader(connectionSocket.getInputStream()));

DataOutputStream outToClient = new DataOutputStream(connectionSocket.getOutputStream());

clientSentence = inFromClient.readLine();

System.out.println("Received: " + clientSentence);

capitalizedSentence = clientSentence.toUpperCase() + '\n';

outToClient.writeBytes(capitalizedSentence);

}

}

}

**CODING (CLIENT)**

import java.io.\*;

import java.net.\*;

class TCPClient{

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

{

String sentence;

String modifiedSentence;

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

Socket clientSocket = new Socket("localhost", 6789);

DataOutputStream outToServer = new DataOutputStream(clientSocket.getOutputStream());

BufferedReader inFromServer = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

sentence = inFromUser.readLine();

outToServer.writeBytes(sentence + '\n');

modifiedSentence = inFromServer.readLine();

System.out.println("FROM SERVER: " + modifiedSentence);

clientSocket.close();

}}

**OUTPUT (SERVER)**

\>java TCPServer

Received: hi

Received: Server

**OUTPUT (CLIENT)**

\>java TCPClient

hi

FROM SERVER: HI

\>java TCPClient

Server

FROM SERVER: SERVER

**FILE PROGRAM FOR READING PALINDROME**

**CODING**

import java.io.\*;

public class PalindromDetector {

public static void main(String[] args) {

try {

FileInputStream fstream = new FileInputStream("d: /Main.txt");

DataInputStream in = new DataInputStream(fstream);

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

String strLine = null;

while ((strLine = br.readLine()) != null) {

String reverse = new

StringBuffer(strLine).reverse().

toString();

int i,j,counter=0;

String m[]=strLine.split(" ");

String[] word=reverse.split(" ");

System.out.println("The palindrome words are:");

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

for(j=word.length-1;j>=0;j--) {

if(m[i].equalsIgnoreCase(word[j])) {

System.out.println(m[i]);

counter++;

break;

}

}

}

System.out.println("Number of palindromes:"+counter);

}

}

catch(IOException e){}

}

}

**FILE(MAIN.TXT)**

Madam Will

Speak

Malayalam

**OUTPUT**

\>java PalindromDetector

The palindrome words are:

Madam

Number of palindromes:1

The palindrome words are:

Number of palindromes:0

The palindrome words are:

Malayalam

Number of palindromes:1

**ECHO SERVER**

**CODING (SERVER)**

import java.io.\*;

import java.net.\*;

public class EchoServer{

public static void main(String [] args)

{

System.out.println("Server Started....");

try

{

ServerSocket ss=new ServerSocket(300);

while(true)

{

Socket s= ss.accept();

Thread t = new ThreadedServer(s);

t.start();

}

}

catch(Exception e)

{

System.out.println("Error: " + e);

}

}

}

class ThreadedServer extends Thread{

Socket soc;

public ThreadedServer(Socket soc)

{

this.soc=soc;

}

public void run()

{

try

{

BufferedReader in=new BufferedReader(new InputStreamReader(soc.getInputStream()));

PrintWriter out=new PrintWriter(soc.getOutputStream());

String str=in.readLine();

System.out.println("Message From Client:"+str);

out.flush();

out.println("Message To Client:"+str);

out.flush();

}

catch(Exception e)

{

System.out.println("Exception:"+e);

}

}

}

**CODING (CLIENT)**

import java.net.\*;

import java.io.\*;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

class EchoClient extends JFrame

{

JTextArea ta;

JTextField msg;

JPanel panel;

JScrollPane scroll;

JButton b1=new JButton("Close");

JButton b2=new JButton("Send");

JLabel l1=new JLabel("Echo Client GUI");

Container c;

EchoClient()

{

c=getContentPane();

setSize(300,470);

setTitle("GUI Client");

panel=new JPanel();

msg=new JTextField(20);

panel.setLayout(new FlowLayout(FlowLayout.CENTER));

ta=new JTextArea(20,20);

scroll=new JScrollPane(ta);

panel.add(l1);

panel.add(ta);

panel.add(msg);

panel.add(b2);

panel.add(b1);

c.add(panel);

b2.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent ae)

{

try

{

Socket s=new Socket("localhost",300);

BufferedReader in=new BufferedReader(new InputStreamReader(s.getInputStream()));

PrintWriter out=new PrintWriter(new OutputStreamWriter(s.getOutputStream()));

out.println(msg.getText());

out.flush();

String temp =ta.getText();

if(temp.equalsIgnoreCase("quit"))

{

System.exit(0);

}

msg.setText("");

ta.append("\n"+in.readLine());

}

catch (IOException e)

{

ta.setText("Exception:"+e);

}

}

});

b1.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

System.exit(0);

}

});

}

public static void main(String args[])

{

EchoClient frame=new EchoClient();

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

frame.setVisible(true);

}

}

**OUTPUT (SERVER)**

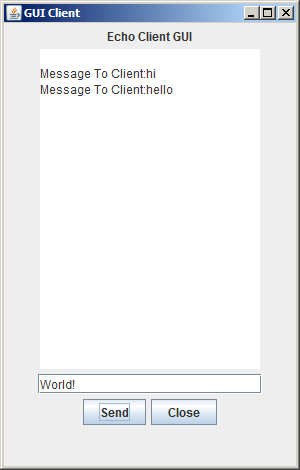
\>java EchoServer

Server Started....

Message From Client:hi

Message From Client:hello

**OUTPUT (CLIENT)**



**JDBC CONNECTION**

**CODING**

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class TestOracleJDBC {

public static void main(String[] args) {

Connection con = null;

Statement stmt = null;

ResultSet rs = null;

try {

Class.forName("oracle.jdbc.driver.OracleDriver");

con = DriverManager.getConnection("jdbc:oracle:thin:@172.16.10.8:1521:kg","14mca","14mca");

stmt = con.createStatement();

rs = stmt.executeQuery("SELECT \* FROM person");

while(rs.next()) {

System.out.print(rs.getInt(1) + "\t");

System.out.println(rs.getString(2));

}

} catch (ClassNotFoundException e) {

e.printStackTrace();

} catch (SQLException e) {

e.printStackTrace();

} finally {

try {

rs.close();

stmt.close();

con.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

}

**OUTPUT**

1 AAA

2 BBB

3 CCC

**NOTEPAD EDITOR**

**CODING**

import java.awt.\*;

import java.awt.event.\*;

import java.io.\*;

class MyNotepade extends Frame implements ActionListener

{

private TextArea txta\_show;

private Menu file;

private MenuItem New,Open,Save,Exit;

private MenuBar mb=new MenuBar();

public MyNotepade()

{

file=new Menu("File");

New=new MenuItem("New");

Open=new MenuItem("Open");

Save=new MenuItem("Save");

Exit=new MenuItem("Exit");

file.add(New);

file.add(Open);

file.add(Save);

file.add(Exit);

mb.add(file);

New.addActionListener(this);

Open.addActionListener(this);

Save.addActionListener(this);

Exit.addActionListener(this);

setTitle("Notpad");

setSize(400,400);

setLocation(100,100);

setMenuBar(mb);

txta\_show=new TextArea();

add(txta\_show);

addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent we)

{

System.exit(0);

}

});

}

public void actionPerformed(ActionEvent ae)

{

if(ae.getSource()==New)

{

txta\_show.setText(" "); }

else if(ae.getSource()==Open)

{

try{

FileDialog fd=new FileDialog(this,"Open File",FileDialog.LOAD);

fd.setVisible(true);

String dir=fd.getDirectory();

String fname=fd.getFile();

FileInputStream fis=new FileInputStream(dir+fname);

DataInputStream dis=new DataInputStream(fis);

String str=" ",msg=" ";

while((str=dis.readLine())!=null)

{

msg=msg+str;

msg+="\n";

}

txta\_show.setText(msg);

dis.close();

}

catch(Exception e){}

}

else if(ae.getSource()==Save)

{

try

{

FileDialog fd=new FileDialog(this,"Save File",FileDialog.SAVE);

fd.setVisible(true);

String txt=txta\_show.getText();

String dir=fd.getDirectory();

String fname=fd.getFile();

FileOutputStream fos=new FileOutputStream(dir+fname);

DataOutputStream dos=new DataOutputStream(fos);

dos.writeBytes(txt);

dos.close();

}

catch(Exception e){}

}

else if(ae.getSource()==Exit)

{

System.exit(0);

}

}

}

class Notepad

{

public static void main(String []args)

{

new MyNotepade().setVisible(true);

}

}

**OUTPUT**



**EDITOR WINDOW**

**CODING**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.io.\*;

public class sw1 extends JFrame implements ActionListener, KeyListener

{

*// A frame with text field and Label field*

JToolBar tb = new JToolBar();

*// icf & b1 an array of values.*

String icf[] = {"new.gif","open.gif","save.gif","cut.gif","copy.gif","paste.gif"};

*// b1 represents for Tooltip*

String bl[] ={"New","Open","Save","Cut","Copy","Paste"};

*// Demonstrate a one-dimensional array*

ImageIcon ic[] = new ImageIcon[icf.length];

*// use a lable to display syntax highlighting*

Label l1 = new Label("Syntax Highlighting");

JButton mbl[] = new JButton[icf.length];

*// set up manu bar*

MenuBar mbar = new MenuBar();

Menu a = new Menu("File");

Menu b = new Menu("Edit");

Menu c = new Menu("Font");

Menu d = new Menu("Font-Size");

Menu f = new Menu("System");

int fsize = 13; *// assigns fsize the Value 13*

JTextArea ta,ta1;

JScrollPane js1;

*// create a fontList using GraphicsEnvironment*

String fontList[] =

GraphicsEnvironment.getLocalGraphicsEnvironment().getAvailableFontFamilyNames();

*// create MenuItem*

MenuItemm[] = new MenuItem[fontList.length];

MenuItemfs[] = new MenuItem[600];

MenuItem a1 = new MenuItem("New");

MenuItem a2 = new MenuItem("Open");

MenuItem a3 = new MenuItem("Save");

MenuItem a4 = new MenuItem("Save as");

MenuItem a5 = new MenuItem("-");

MenuItem a6 = new MenuItem("Exit");

MenuItem b1 = new MenuItem("Cut");

MenuItem b2 = new MenuItem("Copy");

MenuItem b3 = new MenuItem("Paste");

MenuItem d1 = new MenuItem("Compile");

MenuItem d2 = new MenuItem("Run");

String ck="";

int i;

public sw1()

{

super("Java Editor");*//set title*

setSize(1024,768); *//assign a layoutsize*

setLayout(null);

ta=new JTextArea(10,10);

js1=new JScrollPane(ta);

ta1=new JTextArea(5,60);

add(js1);add(ta);add(ta1);add(tb);

a.add(a1);add(l1);

a.add(a2);

a.add(a3);

a.add(a4);

a.add(a5);

a.add(a6);

b.add(b1);

b.add(b2);

b.add(b3);

f.add(d1);

f.add(d2);

mbar.add(a);

mbar.add(b);

mbar.add(c);

mbar.add(d);

mbar.add(f);

setMenuBar(mbar);

*// loop for FontList*

for (i = 0; i < fontList.length; i++ )

{

m[i] = new MenuItem(fontList[i]);

m[i].addActionListener(this);

m[i].setActionCommand(""+i);

c.add(m[i]);

}

*// loop for MenuItem*

for (i = 8; i <=90 ; i+=2)

{

fs[i] = new MenuItem(String.valueOf(i));

fs[i].addActionListener(new fs());

d.add(fs[i]);

}

*//loop for ImageIcone*

for( i =0;i<bl.length;i++)

{

ic[i]= new ImageIcon(icf[i]);

mbl[i] = new JButton(ic[i]);

mbl[i].setToolTipText(bl[i]);

mbl[i].addActionListener(this);

mbl[i].setActionCommand("z"+i);

if(i==3)tb.addSeparator();

tb.add(mbl[i]);

}

tb.reshape(0,0,500,35);

ta.reshape(5,40,1005,500);

js1.reshape(5,4,100,35);

*// set foregroundcolor for l1 that is Syntax Highlighting*

l1.setForeground(Color.green);

*// set foregroundcolor for ta that is text area*

ta.setForeground(Color.blue);

l1.reshape(400,550,130,30);

ta1.reshape(5,600,1005,200);

ta1.setForeground(Color.red);

add(tb);

add(ta);

add(js1);

add(l1);

*// This Listener can performed in wh object*

a6.addWindowListener(new wh());

*// This Listener can be performed in this object*

a1.addActionListener(this);

a2.addActionListener(this);

a3.addActionListener(this);

a4.addActionListener(this);

*// This Listener can be performed in wc object*

a5.addActionListener(new wc());

b1.addActionListener(new wc());

b2.addActionListener(new wc());

b3.addActionListener(new wc());

d1.addActionListener(new wc());

d2.addActionListener(new wc());

ta.addKeyListener(this);

show();

}

*//set button action*

public class wh extends WindowAdapter {

public void windowClosing(WindowEvent e) {

System.exit(0);

}

}

*// an ActionListenerthat sets the FontSize*

public class fs implements ActionListener {

public void actionPerformed(ActionEvent e) {

fsize = Integer.parseInt(e.getActionCommand());

System.out.println(fsize);

}}

public class wc implements ActionListener {

public void actionPerformed(ActionEvent e) {

*// create button actions*

String cmd= e.getActionCommand();

*// associate actions with buttons*

if (cmd.equals("Exit")) System.exit(0);

if (cmd.equals("Cut")) ta.cut();

if (cmd.equals("Copy")) ta.copy();

if (cmd.equals("Paste")) ta.paste();

if (cmd.equals("Save"))

{

}

if (cmd.equals("Compile"))

{

try {

Runtime rt= Runtime.getRuntime();

Process p=rt.exec("taskmgr.exe");

}

catch(Exception ea) {

System.out.println("Error");

}

}

}

}

public void keyReleased(KeyEvent ke){}

public void keyPressed(KeyEvent ke){}

public void keyTyped(KeyEvent ke)

{

*// create a KeyPressed , KeyReleased, Key Typed*

Character k = ke.getKeyChar();

int mk = (int) k;

System.out.println(mk);

ck += k;

*// create KeyReleased action*

if (ck.equals("for "))

{

ta1.setText("");

ta1.append("for(var=init;condition;incr/decr)\n");

ta1.append("{\n");

ta1.append(" set of statements;\n");

ta1.append("}\n");

ck="";

}

if (ck.equals("if "))

{

ta1.setText("");

ta1.append("if (condition)\n");

ta1.append("{\n");

ta1.append(" set of statements;\n");

ta1.append("}\n");

ta1.append("else\n");

ta1.append("{\n");

ta1.append(" set of statements;\n");

ta1.append("}\n");

ck = "";

}

if (ck.equals("do "))

{

ta1.setText("");

ta1.append("do\n");

ta1.append("{\n");

ta1.append(" set of statements;\n");

ta1.append("}\n");

ta1.append("while (condition);\n");

ck = "";

}

if (ck.equals("while "))

{

ta1.setText("");

ta1.append("while(condition)\n");

ta1.append("{\n");

ta1.append(" set of statements;\n");

ta1.append("}\n");

ck = "";

}

if (mk == 10 || mk == 32 || mk == 8 || mk == 9) ck ="";

}

*//to create Button Link*

public void actionPerformed(ActionEvent ae)

{

String s1 = (String) ae.getActionCommand();

int ck1=1;

System.out.println(s1);

if (s1.equals("z0"))

{

ta.setText("");ck1=0;

}

if (s1.equals("z3"))

{

ta.cut();ck1=0;

}

if (s1.equals("z4"))

{

ta.copy();ck1=0;

}

if (s1.equals("z5"))

{

ta.paste();ck1=0;

}

if (s1.equals("z1"))

{

*// for open a filemenu*

FileDialog fx = new FileDialog(this,"Open a File",FileDialog.LOAD);

fx.setDirectory("d:\\jeditor");

fx.setFile("\*.\*");

fx.show();

String mfname=fx.getDirectory()+fx.getFile();

setTitle(mfname);

File fx1=new File(mfname);

try

{

FileInputStream fis=new FileInputStream(fx1);

int fsize=(int) fx1.length();

byte d[] =new byte[fsize];

fis.read(d);

ta.setText("");

ta.setText(new String(d));

}

catch(Exception ex){};

ck1=0;

}

if (s1.equals("z2"))

{

*// for open a savemenu bar*

FileDialog f=new FileDialog(this,"Save a File",FileDialog.SAVE);

f.setDirectory("d:\\jeditor");

f.setFile("\*.java");

f.show();

String fname=f.getDirectory()+f.getFile();

setTitle(fname);

File f1=new File(fname);

try

{

FileOutputStream fos=new FileOutputStream(f1);

String t=ta.getText();

int fsize=t.length();

byte[] d=new byte[fsize];

t.getBytes(0,fsize,d,0);

fos.write(d);

fos.close();

}

catch(Exception e){};

ck1=0;

}

if ((s1.compareTo("Open") != 0)&& (s1.compareTo("Save")) != 0&& (s1.compareTo("New") != 0) && (s1.compareTo("Save as") != 0 && (ck1 == 1)))

{

int x = Integer.parseInt(ae.getActionCommand());

String fname = fontList[x];

Font f1 = new Font(fname,Font.BOLD,fsize);

ta.setFont(f1);

}

if (s1.equals("Open"))

{

*// for open a openmenu bar*

FileDialog fx = new FileDialog(this,"Open a File",FileDialog.LOAD);

fx.setDirectory("d:\\jeditor");

fx.setFile("\*.\*");

fx.show();

String mfname=fx.getDirectory()+fx.getFile();

setTitle(mfname);

File fx1=new File(mfname);

try

{

FileInputStream fis=new FileInputStream(fx1);

int fsize=(int) fx1.length();

byte d[] =new byte[fsize];

fis.read(d);

ta.setText("");

ta.setText(new String(d));

}

catch(Exception ex){};

}

if (s1.equals("New"))

{

ta.setText("");

}

if (s1.equals("Save") ||s1.equals("Save as"))

{

*// for open a save or saveasmenu bar*

FileDialog f=new FileDialog(this,"Save a File",FileDialog.SAVE);

f.setDirectory("d:\\jeditor");

f.setFile("\*.java");

f.show();

String fname=f.getDirectory()+f.getFile();

setTitle(fname);

File f1=new File(fname);

try

{

FileOutputStream fos=new FileOutputStream(f1);

String t=ta.getText();

int fsize=t.length();

byte[] d=new byte[fsize];

t.getBytes(0,fsize,d,0);

fos.write(d);

fos.close();

}

catch(Exception e){};

}

}

public static void main(String args[])

{

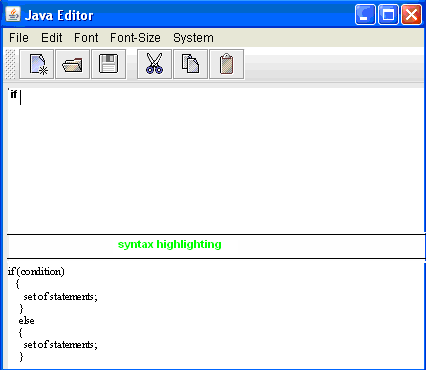
sw1 app = new sw1();

}

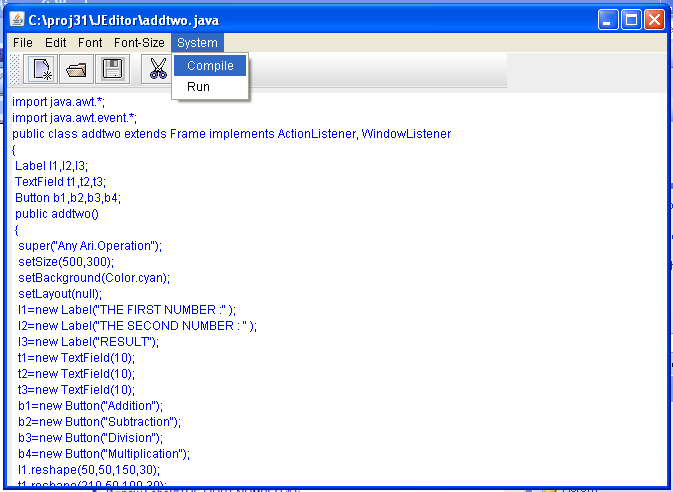
}

**OUTPUT**

**HIGHLIGHTING SYNTAX**



**COMPILING**



**STUDENT DETAILS**

**CODING**

import java.sql.\*;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.util.\*;

import javax.swing.table.\*;

import javax.swing.border.\*;

class StudDet

{

public static void main(String args[])

{

new School();

}

}

class School extends JFrame

{

JFrame frame;

JTabbedPane tab;

School()

{

frame=new JFrame("School Management System");

frame.setSize(500,500);

frame.setVisible(true);

frame.setDefaultCloseOperation(3);

tab=new JTabbedPane();

frame.add(tab,BorderLayout.CENTER);

Student st=new Student();

tab.add("Student",st);

Teacher teach=new Teacher();

tab.add("Teacher",teach);

View view=new View();

tab.add("Reports",view);

}

}

class Student extends JPanel implements ActionListener

{

JLabel lname,lgen,ldob,lcourse,lyear,lmobile,ltitle;

JTextField tname,tdob,tcourse,tmobile;

JComboBox cyear;

JRadioButton rm,rfm;

JButton bsubmit,bcancel;

ButtonGroup group;

Student()

{

setLayout(null);

ltitle=new JLabel("STUDENTS REGISTRATIONS");

lname=new JLabel("Name ");

lgen=new JLabel("Gender ");

ldob=new JLabel("Dob ");

lcourse=new JLabel("Course ");

lyear=new JLabel("Year ");

lmobile=new JLabel("Mobile ");

tname=new JTextField();

tdob=new JTextField();

tcourse=new JTextField();

tmobile=new JTextField();

cyear=new JComboBox();

cyear.addItem("I-year");

cyear.addItem("II-year");

cyear.addItem("III-year");

rm=new JRadioButton("Male");

rfm=new JRadioButton("Female");

group=new ButtonGroup();

group.add(rm);

group.add(rfm);

bsubmit=new JButton("Submit");

bcancel=new JButton("Cancel");

ltitle.setBounds(140,10,250,25);

lname.setBounds(100,50,70,25);

lgen.setBounds(100,80,70,25);

ldob.setBounds(100,110,70,25);

lcourse.setBounds(100,140,70,25);

lyear.setBounds(100,170,70,25);

lmobile.setBounds(100,200,70,25);

tname.setBounds(170,50,150,25);

rm.setBounds(170,80,70,25);

rfm.setBounds(240,80,80,25);

tdob.setBounds(170,110,150,25);

tcourse.setBounds(170,140,150,25);

cyear.setBounds(170,170,150,25);

tmobile.setBounds(170,200,150,25);

bsubmit.setBounds(120,240,80,25);

bcancel.setBounds(205,240,80,25);

add(ltitle);

add(lname);

add(tname);

add(lgen);

add(rm);add(rfm);

add(ldob);

add(tdob);

add(lcourse);

add(tcourse);

add(lyear);

add(cyear);

add(lmobile);

add(tmobile);

add(bsubmit);

add(bcancel);

bsubmit.addActionListener(this);

bcancel.addActionListener(this);

}

public void actionPerformed(ActionEvent ae)

{

if(ae.getSource().equals(bsubmit))

{

Connection con;

Statement st;

ResultSet rs;

String nm="",gen="",dob="",crs="",year="",mob="";

nm=tname.getText();

if(rm.isSelected())

gen=rm.getText();

else if(rfm.isSelected())

gen=rfm.getText();

dob=tdob.getText();

crs=tcourse.getText();

year=(String)cyear.getSelectedItem();

mob=tmobile.getText();

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

con=DriverManager.getConnection("jdbc:oracle:thin:@127.0.0.1:1521:orcl","mca","mca");

st=con.createStatement();

st.executeUpdate("insert into tb\_std values('"+nm+"','"+gen+"','"+dob+"','"+crs+"','"+year+"','"+mob+"')");

JOptionPane.showMessageDialog(null,"Student Registered Successfully !");

}

catch(Exception exp){}

}//bsubmit

if(ae.getSource().equals(bcancel))

{

tname.setText("");

rm.setSelected(false);

rfm.setSelected(false);

tdob.setText("");

tcourse.setText("");

tmobile.setText("");

tname.requestFocus();

}//bcancel

}

}

class Teacher extends JPanel implements ActionListener

{

JLabel lname,lgen,ldob,lqlf,lexp,lmobile,ltitle;

JTextField tname,tdob,tqlf,texp,tmobile;

JRadioButton rm,rfm;

JButton bsubmit,bcancel;

ButtonGroup group;

Teacher()

{

setLayout(null);

ltitle=new JLabel("TEACHERS REGISTRATIONS");

lname=new JLabel("Name ");

lgen=new JLabel("Gender ");

ldob=new JLabel("Dob ");

lqlf=new JLabel("Qualification ");

lexp=new JLabel("Experience ");

lmobile=new JLabel("Mobile ");

tname=new JTextField();

tdob=new JTextField();

tqlf=new JTextField();

texp=new JTextField();

tmobile=new JTextField();

rm=new JRadioButton("Male");

rfm=new JRadioButton("Female");

group=new ButtonGroup();

group.add(rm);

group.add(rfm);

bsubmit=new JButton("Submit");

bcancel=new JButton("Cancel");

ltitle.setBounds(140,10,250,25);

lname.setBounds(90,50,90,25);

lgen.setBounds(90,80,90,25);

ldob.setBounds(90,110,90,25);

lqlf.setBounds(90,140,90,25);

lexp.setBounds(90,170,90,25);

lmobile.setBounds(90,200,90,25);

tname.setBounds(170,50,150,25);

rm.setBounds(170,80,70,25);

rfm.setBounds(240,80,80,25);

tdob.setBounds(170,110,150,25);

tqlf.setBounds(170,140,150,25);

texp.setBounds(170,170,150,25);

tmobile.setBounds(170,200,150,25);

bsubmit.setBounds(120,240,80,25);

bcancel.setBounds(205,240,80,25);

add(ltitle);

add(lname);

add(tname);

add(lgen);

add(rm);add(rfm);

add(ldob);

add(tdob);

add(lqlf);

add(tqlf);

add(lexp);

add(texp);

add(lmobile);

add(tmobile);

add(bsubmit);

add(bcancel);

bsubmit.addActionListener(this);

bcancel.addActionListener(this);

}

public void actionPerformed(ActionEvent ae)

{

if(ae.getSource().equals(bsubmit))

{

Connection con;

Statement st;

ResultSet rs;

String nm="",gen="",dob="",qlf="",exp="",mob="";

nm=tname.getText();

if(rm.isSelected())

gen=rm.getText();

else if(rfm.isSelected())

gen=rfm.getText();

dob=tdob.getText();

qlf=tqlf.getText();

exp=texp.getText();

mob=tmobile.getText();

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

con=DriverManager.getConnection("jdbc:oracle:thin:@127.0.0.1:1521:orcl","mca","mca");

st=con.createStatement();

st.executeUpdate("insert into tb\_teach values('"+nm+"','"+gen+"','"+dob+"','"+qlf+"','"+exp+"','"+mob+"')");

JOptionPane.showMessageDialog(null,"Student Registered Successfully !");

}

catch(Exception ee){}

}//bsubmit

if(ae.getSource().equals(bcancel))

{

tname.setText("");

rm.setSelected(false);

rfm.setSelected(false);

tdob.setText("");

tqlf.setText("");

tmobile.setText("");

texp.setText("");

tname.requestFocus();

}//bcancel

}

}

class View extends JPanel

{

JTable table;

Vector<Vector> rows=new Vector<Vector>();

Vector<String> row=new Vector<String>();

Vector<String> cols=new Vector<String>();

View()

{

setLayout(new BorderLayout());

cols.add("SNAME");

cols.add("GENDER");

cols.add("DOB");

cols.add("COUSRE");

cols.add("YEAR");

cols.add("MOBILE");

Connection con;

Statement st;

ResultSet rs;

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

con=DriverManager.getConnection("jdbc:oracle:thin:@127.0.0.1:1521:orcl","mca","mca");

st=con.createStatement();

rs=st.executeQuery("select \* from tb\_std");

while(rs.next())

{

row.add(rs.getString("sname"));

row.add(rs.getString("gender"));

row.add(rs.getString("dob"));

row.add(rs.getString("course"));

row.add(rs.getString("year"));

row.add(rs.getString("mobile"));

rows.add(row);

}

}

catch(Exception exp){}

table=new JTable(rows,cols);

add(table);

table.setRowHeight(20);

table.setGridColor(Color.red);

table.setBorder(BorderFactory.createLineBorder(Color.black,3));

JTableHeader head=table.getTableHeader();

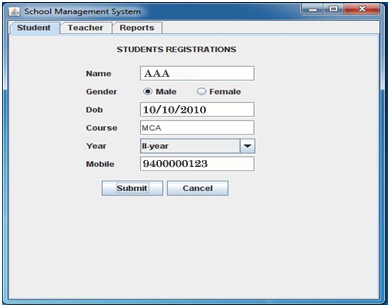
add(head,BorderLayout.NORTH);

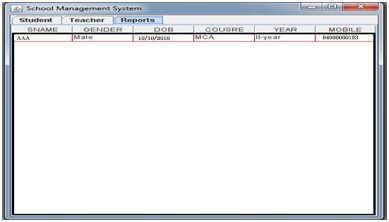
add(table,BorderLayout.CENTER);

}

}

**OUTPUT**

****

****

**BANK PROCESS**

**CODING**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.sql.\*;

import java.text.\*;

import java.util.Date;

class Login extends JFrame implements ActionListener

{

JFrame frame;

JPanel p1,p2,p3;

JTextField tuname;

JLabel[] llogin=new JLabel[3];

JButton blogin,bnew;

JPasswordField pwd;

Connection con;

Statement st;

ResultSet rs;

Login()

{

frame=new JFrame();

frame.setTitle("Login Process");

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

frame.show();

frame.setResizable(false);

frame.setSize(300,300);

frame.setLayout(null);

llogin[0]=new JLabel("Login Here");

llogin[1]=new JLabel("Account No :");

llogin[2]=new JLabel("Password :");

tuname=new JTextField();

pwd=new JPasswordField();

blogin=new JButton("Login");

bnew=new JButton("New Account Creation");

llogin[0].setBounds(100,20,100,25);

llogin[1].setBounds(20,70,100,25);

llogin[2].setBounds(20,110,100,25);

tuname.setBounds(100,70,150,25);

pwd.setBounds(100,110,150,25);

blogin.setBounds(180,150,70,25);

bnew.setBounds(80,200,170,20);

frame.add(llogin[0]);

frame.add(llogin[1]);

frame.add(llogin[2]);

frame.add(tuname);

frame.add(pwd);

frame.add(blogin);

frame.add(bnew);

blogin.addActionListener(this);

bnew.addActionListener(this);

tuname.setText(""+0);

}//Login Constructor

public void actionPerformed(ActionEvent ae)

{

if(ae.getSource().equals(blogin))

{

try

{

int un=Integer.parseInt(tuname.getText());

String pwd1=pwd.getText();

if(un!=0){

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

con=DriverManager.getConnection("jdbc:odbc:dsnreg","mca","mca");

st=con.createStatement();

rs=st.executeQuery("select \* from lreg where acno='"+un+"' and password='"+pwd1+"' ");

if(rs.next()){

if(rs.getInt("acno")==un &&rs.getString("password").equals(pwd1)){

Transaction t=new Transaction();}

}

else

JOptionPane.showMessageDialog(null,"Invalid Password !");

}

else

JOptionPane.showMessageDialog(null,"Please Enter the Account Number");

}catch(Exception exp){}

}

if(ae.getSource().equals(bnew))

{

Register reg=new Register();

}

}

}//Login Class

class Register extends JFrame implements ActionListener

{

JFrame frame;

JTextField[] treg=new JTextField[10];

JLabel[] lreg=new JLabel[10];

JLabel ltitle,ldate,ldate1;

JPasswordField tpwd;

JButton bsubmit,breset;

JComboBox cmb;

Connection con;

Statement st;

ResultSet rs;

SimpleDateFormat sdf=new SimpleDateFormat("dd-MM-yyyy");

Date dt=new Date();

int acno=101;

Register()

{

frame=new JFrame();

frame.setTitle("Registration");

frame.show();

frame.setSize(370,400);

frame.setLayout(null);

frame.setResizable(false);

ltitle=new JLabel("Account Opening Form");

ldate=new JLabel("Date :");

ldate1=new JLabel();

ldate1.setText(sdf.format(dt));

lreg[0]=new JLabel("Account No");

lreg[1]=new JLabel("Name ");

lreg[2]=new JLabel("Password ");

lreg[3]=new JLabel("DateOfBirth");

lreg[4]=new JLabel("Gender ");

lreg[5]=new JLabel("City ");

lreg[6]=new JLabel("Mobile No ");

lreg[7]=new JLabel("Email ID ");

treg[0]=new JTextField();

treg[1]=new JTextField();

tpwd=new JPasswordField();

cmb=new JComboBox();

cmb.addItem("--select--");

cmb.addItem("Male");

cmb.addItem("Female");

treg[2]=new JTextField();

treg[3]=new JTextField();

treg[4]=new JTextField();

treg[5]=new JTextField();

bsubmit=new JButton("Submit");

breset=new JButton("Reset");

ltitle.setBounds(80,10,200,30);

ldate.setBounds(220,40,70,25);

ldate1.setBounds(250,40,80,25);

lreg[0].setBounds(20,75,100,25);

lreg[1].setBounds(20,100,100,25);

lreg[2].setBounds(20,125,100,25);

lreg[3].setBounds(20,150,100,25);

lreg[4].setBounds(20,175,100,25);

lreg[5].setBounds(20,200,100,25);

lreg[6].setBounds(20,225,100,25);

lreg[7].setBounds(20,250,100,25);

treg[0].setBounds(120,75,200,25);

treg[1].setBounds(120,100,200,25);

tpwd.setBounds(120,125,200,25);

treg[2].setBounds(120,150,200,25);

cmb.setBounds(120,175,200,25);

treg[3].setBounds(120,200,200,25);

treg[4].setBounds(120,225,200,25);

treg[5].setBounds(120,250,200,25);

bsubmit.setBounds(130,300,80,25);

breset.setBounds(220,300,80,25);

frame.add(ltitle);

frame.add(ldate);

frame.add(ldate1);

frame.add(lreg[0]);

frame.add(lreg[1]);

frame.add(lreg[2]);

frame.add(lreg[3]);

frame.add(lreg[4]);

frame.add(lreg[5]);

frame.add(lreg[6]);

frame.add(lreg[7]);

frame.add(treg[0]);

frame.add(treg[1]);

frame.add(tpwd);

frame.add(cmb);

frame.add(treg[2]);

frame.add(treg[3]);

frame.add(treg[4]);

frame.add(treg[5]);

frame.add(bsubmit);

frame.add(breset);

bsubmit.addActionListener(this);

breset.addActionListener(this);

try{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

con=DriverManager.getConnection("jdbc:odbc:dsnreg","mca","mca");

st=con.createStatement();

rs=st.executeQuery("select \* from lreg order by acno asc");

while(rs.next())

acno=rs.getInt("acno");

acno=acno+1;

if(acno<=101)

acno=101;

treg[0].setText(""+acno);

}catch(Exception exp){}

}//Register Constructor

public void actionPerformed(ActionEvent ae)

{

if(ae.getSource().equals(bsubmit))

{

String name="",pwd="",dob="",gender="",city="",mobile="",mail="",doc="";

name=treg[1].getText();

pwd=tpwd.getText();

dob=treg[2].getText();

gender=(String)cmb.getSelectedItem();

city=treg[3].getText();

mobile=treg[4].getText();

mail=treg[5].getText();

doc=ldate1.getText();

try{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

con=DriverManager.getConnection("jdbc:odbc:dsnreg","mca","mca");

st=con.createStatement();

st.executeUpdate("insert into lreg(acno,name,password,dob,gender,city,mobile,email,dateoc) values('"+acno+"','"+name+"','"+pwd+"','"+dob+"','"+gender+"','"+city+"','"+mobile+"','"+mail+"','"+doc+"')");

con.close();

JOptionPane.showMessageDialog(null,"Registered Successfully !");

frame.dispose();

}catch(Exception exp){}

}

if(ae.getSource().equals(breset))

{

treg[0].setText("");

treg[1].setText("");

treg[2].setText("");

treg[3].setText("");

treg[4].setText("");

treg[5].setText("");

cmb.setSelectedIndex(0);

tpwd.setText("");

}

}

}//Register Class

class Transaction extends JFrame implements ActionListener

{

JFrame frame;

JLabel ltype,ltitle,lavail,lbal,lacno,lpwd,lname,ldate,ldate1;

JTextField tano,tpwd,tname,tacno,tavail,tbal;

JComboBox cmb;

JButton bview,bok,bcancel;

Connection con;

Statement st;

ResultSet rs;

Transaction()

{

frame=new JFrame();

frame.setTitle("Transaction");

frame.show();

frame.setSize(320,350);

frame.setLayout(null);

frame.setResizable(false);

ltitle=new JLabel("WELCOME INDIAN BANK");

ltype=new JLabel("Select Transaction :");

lacno=new JLabel("Enter Account No :");

lpwd=new JLabel("Enter Password :");

lname=new JLabel("Customer Name :");

lavail=new JLabel("Available Balance:");

lbal=new JLabel("Enter the Amount :");

cmb=new JComboBox();

cmb.addItem("--select--");

cmb.addItem("Deposit");

cmb.addItem("Withdraw");

tano=new JTextField();

tpwd=new JPasswordField();

tname=new JTextField();

tavail=new JTextField();

tbal=new JTextField();

bview=new JButton("View");

bok=new JButton("Ok");

bcancel=new JButton("Check");

ltitle.setBounds(80,20,200,25);

ltype.setBounds(20,60,130,25);

lacno.setBounds(20,90,130,25);

lpwd.setBounds(20,120,130,25);

lname.setBounds(20,150,130,25);

lavail.setBounds(20,180,130,25);

lbal.setBounds(20,210,130,25);

cmb.setBounds(150,60,100,25);

tano.setBounds(150,90,150,25);

tpwd.setBounds(150,120,150,25);

tname.setBounds(150,150,150,25);

tavail.setBounds(150,180,150,25);

tbal.setBounds(150,210,150,25);

bview.setBounds(70,260,70,25);

bok.setBounds(150,260,60,25);

bcancel.setBounds(220,260,80,25);

frame.add(ltitle);

frame.add(ltype);

frame.add(lacno);

frame.add(lpwd);

frame.add(lname);

frame.add(lavail);

frame.add(lbal);

frame.add(cmb);

frame.add(tano);

frame.add(tpwd);

frame.add(tname);

frame.add(tavail);

frame.add(tbal);

frame.add(bview);

frame.add(bok);

frame.add(bcancel);

tavail.setEditable(false);

tano.setText(""+0);

bok.setEnabled(false);

bview.addActionListener(this);

bok.addActionListener(this);

bcancel.addActionListener(this);

}//Constructor

public void actionPerformed(ActionEvent ae)

{

if(ae.getSource().equals(bcancel))

{

int check=0;

try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

con=DriverManager.getConnection("jdbc:odbc:dsnreg","mca","mca");

st=con.createStatement();

int ano=Integer.parseInt(tano.getText());

String pwd1=tpwd.getText();

if(pwd1.equals("")||ano<0)

{

JOptionPane.showMessageDialog(null,"Please enter the Account number and Password");

}

else

{

rs=st.executeQuery("select \* from lreg where acno='"+ano+"' and password='"+pwd1+"' ");

while(rs.next()){

check=1;

tname.setText(""+rs.getString("name"));

tavail.setText(""+rs.getInt("avail"));

bok.setEnabled(true);

}

if(check==0)

JOptionPane.showMessageDialog(null,"Invaild Password");

}

}catch(Exception exp){}

}

if(ae.getSource().equals(bok))

{

if(cmb.getSelectedItem().equals("Withdraw"))

{

try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

con=DriverManager.getConnection("jdbc:odbc:dsnreg","mca","mca");

st=con.createStatement();

int ano=Integer.parseInt(tano.getText());

int avl,amt;

avl=Integer.parseInt(tavail.getText());

amt=Integer.parseInt(tbal.getText());

if(amt<=avl)

{

st.executeUpdate("update lreg set avail='"+(avl-amt)+"' where acno='"+ano+"' ");

JOptionPane.showMessageDialog(null,"Your account has been credited Rs."+amt);

JOptionPane.showMessageDialog(null,"The Remaining Balance Rs."+(avl-amt));

tavail.setText(""+(avl-amt));

tbal.setText(""+0);

}

else

JOptionPane.showMessageDialog(null,"Insufficient Balance of your Account");

}catch(Exception exp){}

}//withdraw

if(cmb.getSelectedItem().equals("Deposit"))

{

try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

con=DriverManager.getConnection("jdbc:odbc:dsnreg","mca","mca");

st=con.createStatement();

int ano=Integer.parseInt(tano.getText());

int avl,amt;

avl=Integer.parseInt(tavail.getText());

amt=Integer.parseInt(tbal.getText());

st.executeUpdate("update lreg set avail='"+(avl+amt)+"' where acno='"+ano+"' ");

JOptionPane.showMessageDialog(null,"Your account has been Debited Rs."+amt);

JOptionPane.showMessageDialog(null,"The Total Balance Rs."+(avl+amt));

tavail.setText(""+(avl+amt));

tbal.setText(""+0);

}catch(Exception exp){}

}

bok.setEnabled(false);

}//ok

if(ae.getSource().equals(bview))

{

Reports r=new Reports();

}

}//ActionListener

class Reports extends JFrame

{

JFrame frame;

JTextField[] treg=new JTextField[10];

JLabel[] lreg=new JLabel[10];

JLabel ltitle,ldate,ldate1;

JPasswordField tpwd;

Connection con;

Statement st;

ResultSet rs;

SimpleDateFormat sdf=new SimpleDateFormat("dd-MM-yyyy");

Date dt=new Date();

int acno=101;

Reports()

{

frame=new JFrame();

frame.setTitle("Reports");

frame.show();

frame.setSize(370,400);

frame.setLayout(null);

frame.setResizable(false);

ltitle=new JLabel("Welcome to Indian Bank");

ldate=new JLabel("Date :");

ldate1=new JLabel();

ldate1.setText(sdf.format(dt));

lreg[0]=new JLabel("Account No");

lreg[1]=new JLabel("Name ");

lreg[2]=new JLabel("Password ");

lreg[3]=new JLabel("DateOfBirth");

lreg[4]=new JLabel("Gender ");

lreg[5]=new JLabel("City ");

lreg[6]=new JLabel("Mobile No ");

lreg[7]=new JLabel("Email ID ");

lreg[8]=new JLabel("Avail Bal ");

lreg[9]=new JLabel("Date Of Open ");

treg[0]=new JTextField();

treg[1]=new JTextField();

treg[2]=new JTextField();

treg[3]=new JTextField();

treg[4]=new JTextField();

treg[5]=new JTextField();

treg[6]=new JTextField();

treg[7]=new JTextField();

treg[8]=new JTextField();

treg[9]=new JTextField();

ltitle.setBounds(80,10,250,30);

ldate.setBounds(220,40,70,25);

ldate1.setBounds(250,40,80,25);

lreg[0].setBounds(20,75,100,25);

lreg[1].setBounds(20,100,100,25);

lreg[2].setBounds(20,125,100,25);

lreg[3].setBounds(20,150,100,25);

lreg[4].setBounds(20,175,100,25);

lreg[5].setBounds(20,200,100,25);

lreg[6].setBounds(20,225,100,25);

lreg[7].setBounds(20,250,100,25);

lreg[8].setBounds(20,275,100,25);

lreg[9].setBounds(20,300,100,25);

treg[0].setBounds(120,75,200,25);

treg[1].setBounds(120,100,200,25);

treg[2].setBounds(120,125,200,25);

treg[3].setBounds(120,150,200,25);

treg[4].setBounds(120,175,200,25);

treg[5].setBounds(120,200,200,25);

treg[6].setBounds(120,225,200,25);

treg[7].setBounds(120,250,200,25);

treg[8].setBounds(120,275,200,25);

treg[9].setBounds(120,300,200,25);

frame.add(ltitle);

frame.add(ldate);

frame.add(ldate1);

frame.add(lreg[0]);

frame.add(lreg[1]);

frame.add(lreg[2]);

frame.add(lreg[3]);

frame.add(lreg[4]);

frame.add(lreg[5]);

frame.add(lreg[6]);

frame.add(lreg[7]);

frame.add(lreg[8]);

frame.add(lreg[9]);

frame.add(treg[0]);

frame.add(treg[1]);

frame.add(treg[2]);

frame.add(treg[3]);

frame.add(treg[4]);

frame.add(treg[5]);

frame.add(treg[6]);

frame.add(treg[7]);

frame.add(treg[8]);

frame.add(treg[9]);

try{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

con=DriverManager.getConnection("jdbc:odbc:dsnreg","mca","mca");

st=con.createStatement();

int a=Integer.parseInt(tano.getText());

rs=st.executeQuery("select \* from lreg where acno='"+a+"' ");

rs.next();

treg[0].setText(""+a);

treg[1].setText(rs.getString("name"));

treg[2].setText(rs.getString("password"));

treg[3].setText(rs.getString("dob"));

treg[4].setText(rs.getString("gender"));

treg[5].setText(rs.getString("city"));

treg[6].setText(rs.getString("mobile"));

treg[7].setText(rs.getString("email"));

treg[8].setText(""+rs.getInt("avail"));

treg[9].setText(rs.getString("dateoc"));

}catch(Exception exp){}

}

}//Reports

}//Transaction Class

class BnkPrc

{

public static void main(String args[])

{

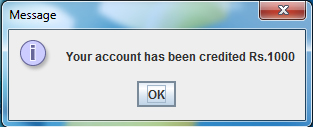
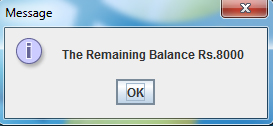
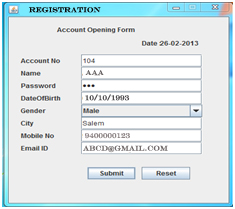
Login login=new Login();

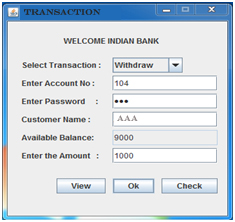
}

}

**OUTPUT**

****

****

****

**OPAC SYSTEM**

**CODING**

import java.sql.\*;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import javax.swing.table.\*;

public class OpacSystem implements ActionListener

{

JRadioButton author=new JRadioButton("Search By Author");

JRadioButton book=new JRadioButton("Search by Book");

JTextField txt=new JTextField(30);

JLabel label=new JLabel("Enter Search Key");

JButton search=new JButton("SEARCH");

JFrame frame=new JFrame();

JTable table;

DefaultTableModel model;

String query="select\*from opacTab";

public OpacSystem()

{

frame.setTitle("OPAC SYSTEM");

frame.setSize(800,500);

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

frame.setLayout(new BorderLayout());

JPanel p1=new JPanel();

p1.setLayout(new FlowLayout());

p1.add(label);

p1.add(txt);

ButtonGroup bg=new ButtonGroup();

bg.add(author);

bg.add(book);

JPanel p2=new JPanel();

p2.setLayout(new FlowLayout());

p2.add(author);

p2.add(book);

p2.add(search);

search.addActionListener(this);

JPanel p3=new JPanel();

p3.setLayout(new BorderLayout());

p3.add(p1,BorderLayout.NORTH);

p3.add(p2,BorderLayout.CENTER);

frame.add(p3,BorderLayout.NORTH);

addTable(query);

frame.setVisible(true);

}

public void addTable(String str)

{

try{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection con=DriverManager.getConnection("jdbc:odbc:opacDS");

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery(str);

ResultSetMetaData rsmd=rs.getMetaData();

int cols=rsmd.getColumnCount();

model=new DefaultTableModel(1,cols);

table=new JTable(model);

String[] tabledata=new String[cols];

int i=0;

while(i<cols)

{

tabledata[i]=rsmd.getColumnName(i+1);

i++;

}

model.addRow(tabledata);

while(rs.next())

{

for(i=0;i<cols;i++)

tabledata[i]=rs.getObject(i+1).toString();

model.addRow(tabledata);

}

frame.add(table,BorderLayout.CENTER);

con.close();

}

catch(Exception e)

{

System.out.println("Exception:"+e);

}

}

public void actionPerformed(ActionEvent ae)

{

if(author.isSelected())

query="select\*from opacTab where AUTHOR like '"+txt.getText()+"%'";

if(book.isSelected())

query="select\*from opacTab where BOOK like '"+txt.getText()+"%'";

while(model.getRowCount()>0)

model.removeRow(0);

frame.remove(table);

addTable(query);

}

public static void main(String[] args)

{

OpacSystem os=new OpacSystem();

}

}

**OUTPUT**

