**JAVA LAB FILE**

List of Programs: --

[1. Program for factorial of an entered number 4](#_Toc278736190)

[2. Program to find the sum of all odd no. Lower than the input no. 4](#_Toc278736191)

[3. WAP using (a) default constructor (b) argument constructor(c) copy constructor 4](#_Toc278736192)

[4. Program to solve the linear Equation 5](#_Toc278736193)

[5. Program to solve a quadratic Equation 5](#_Toc278736194)

[6. To find the volume and surface area of a cuboid. 5](#_Toc278736195)

[7. Program for illustrate a copy constructor so that a string may be duplicated in to another 5](#_Toc278736196)

[8. Program for constructor overloading 6](#_Toc278736197)

[9. WAP to implement in how many ways a team of 11 player out of 15 7](#_Toc278736198)

[10. Wap to calculate the distance travelled by a vehicle by s = ut +(1/2\*(a\*(t\*t)) 7](#_Toc278736199)

[11. Program to implment public private & protected mechanism 7](#_Toc278736200)

[12. WAP for user defined package and interface in which you can write your own interface and classes. 8](#_Toc278736201)

[13 Program to display the records of Books 8](#_Toc278736202)

[14. Program to find armstrong number between 100-1000 9](#_Toc278736203)

[15. Program to implement shift operator 9](#_Toc278736204)

[16. Program to show synchronization between threads 10](#_Toc278736205)

[17. WAP todemonstrate few types of exception using try and catch blocks 10](#_Toc278736206)

[18. WAP to illustrate the use of throw clause 10](#_Toc278736207)

[19. W.A.P to design a calculator 10](#_Toc278736208)

[20. Program to find power of 2 12](#_Toc278736209)

[21. WAP to demonstrate the Mouse handling events. 12](#_Toc278736210)

[22. WAP in awt to demonstrate frame, checkbox, button ,menu of your own choice 14](#_Toc278736211)

[23. Swap two values without using any third variable 15](#_Toc278736212)

[24. Program to change cms in inch and feet 15](#_Toc278736213)

[25. wap for the string manipulation 15](#_Toc278736214)

[26. Program to implement shift opeartors 17](#_Toc278736215)

[27. Program to search an element in an array 17](#_Toc278736216)

[28. Program to solve linear & quadratic equation 18](#_Toc278736217)

[29. WAP using default argument constructor 19](#_Toc278736218)

[30 . WAP to compute whether the given no. is prime or not 19](#_Toc278736219)

[31. Program to find the power of 2 without arithematic operation 19](#_Toc278736220)

[32. Program to find permutation p(n,r)=n!/(n-r)! 19](#_Toc278736221)

[33. Program to find the transpose of a Matrix 20](#_Toc278736222)

[34. Program to display the faculy Records 21](#_Toc278736223)

[35. Program for constructor overloading 22](#_Toc278736224)

[36. Program to demonstrate linked list 22](#_Toc278736225)

[37. Solving a linear equation 23](#_Toc278736226)

[38. Program to implement String Tokenizer 23](#_Toc278736227)

[39. Program to implement final, static and abstract modifier 24](#_Toc278736228)

[40. Program for implementing Fibonacci series 24](#_Toc278736229)

[41. Write a program of ur own which implements interface 24](#_Toc278736230)

[42. Write a program to show at least 8 exceptions. 25](#_Toc278736231)

[43. Program to compute division of 2 without using any mathmatical operator 26](#_Toc278736232)

[44. Program to implement Runtime polymorphism 27](#_Toc278736233)

[45. Program for a frog to reach the top of wall. 27](#_Toc278736234)

[46. WAP to find the sum of Digits 28](#_Toc278736235)

[47. Program to find the combination 15C11 28](#_Toc278736236)

[48. Program to implement public, private and protected mechanism 28](#_Toc278736237)

[49. Program to check whether the given character is a vowel or not 29](#_Toc278736238)

[50. W.A.P to create an applet with button 29](#_Toc278736239)

[51. Program to display actor's name in checkboxes 30](#_Toc278736240)

[52. Program to demonstrate the User-Defined Exceptions. 31](#_Toc278736241)

[53. Program to implement bitwise operators 32](#_Toc278736242)

[54. WAP to create an applet to receive text from textfield and display on applet 33](#_Toc278736243)

[55. Program to create Menu in a Frame. 33](#_Toc278736244)

[56. Program to implement Vector class 33](#_Toc278736245)

[58. Program to implement type casting 34](#_Toc278736246)

[59. Program to display car in list 35](#_Toc278736247)

[60. Program to show magic square in table 35](#_Toc278736248)

[61. Program to create a thread by implementing runnable interface 36](#_Toc278736249)

[62. Program to show synchronization between threads 36](#_Toc278736250)

# 1. Program for factorial of an entered number

import java.util.\*;

public class factorial {

public static void main(String args[]) {

Scanner ob = new Scanner(System.in);

System.out.print("Enter the Number Whose Factorial is to be Calculated = ");

int n = ob.nextInt();

int fact = 1;

for(int i = n ; i > 0 ; i-- )

fact = fact \* i ;

System.out.println("Factorial is = "+fact);

}

}

# 2. Program to find the sum of all odd no. Lower than the input no.

import java.util.Scanner;

public class SUM {

public static void main(String args[]) {

System.out.print("Enter the Number = ");

Scanner in = new Scanner(System.in);

int n = in.nextInt();

int sum = 0;

for(int i= 1 ; i < n ; i+=2)

sum = sum + i;

System.out.println("Sum is = "+sum);

}

}

# 3. WAP using (a) default constructor (b) argument constructor(c) copy constructor

public class Constructor {

int a, b;

Constructor() {

a = 10;

b = 20;

}

Constructor(int a, int b) {

this.a = a;

this.b = b;

}

Constructor(Constructor ob) {

a = ob.a;

b = ob.b;

}

public void show() {

System.out.println("a = " + a + "\nb = " + b);

}

public static void main(String args[]) {

Constructor c = new Constructor();

Constructor c2 = new Constructor(c);

c.show();

c2.show();

}}

OUTPUT

a = 10

b = 20

c = 10

b = 20

# 4. Program to solve the linear Equation

//done in question 28

# 5. Program to solve a quadratic Equation

//done in question 28

# 6. To find the volume and surface area of a cuboid.

import java.util.\*;

public class box {

public static void main(String args[]) {

int l, b, h;

Scanner ob = new Scanner(System.in);

System.out.println("Enter the value of cuboid:lenght,breath,height");

l = ob.nextInt();

b = ob.nextInt();

h = ob.nextInt();

float vol;

vol = l \* b \* h;

System.out.println("volume of cuboid =" + vol);

int area;

area = 2 \* (l \* h + h \* b + b \* l);

System.out.println("area of cuboide =" + area);

}

}

Output:

Enter the value of cuboid:lenght,breath,height

2

32

43

volume of cuboid =2752.0

area of cuboid= 3052

# 7. Program for illustrate a copy constructor so that a string may be duplicated in to another

public class Constructor {

String str;

Constructor() {

str = "hello java";

}

Constructor(String str) {

this.str = str;

}

Constructor(Constructor ob) {

this.str = ob.str;

}

public void show() {

System.out.println(str);

}

public static void main(String args[]) {

Constructor c = new Constructor();

Constructor c2 = new Constructor("java for u");

Constructor c3 = new Constructor(c2);

c.show();

c2.show();

c3.show();

}

}

Output:

hello java

java for u

java for u

# 8. Program for constructor overloading

class Box {

double width;

double height;

double depth;

Box(double w, double h, double d) {

width = w;

height = h;

depth = d;

}

Box() {

width = -1; // use -1 to indicate

height = -1; // an uninitialized

depth = -1; // box

}

Box(double len) {

width = height = depth = len;

}

// compute and return volume

double volume() {

return width \* height \* depth;

}}

class ConstructorOverloading {

public static void main(String args[]) {

// create boxes using the various constructors\

Box mybox1 = new Box(10, 20, 15);

Box mybox2 = new Box();

Box mycube = new Box(7);

double vol;

// get volume of first box

vol = mybox1.volume();

System.out.println("Volume of mybox1 is " + vol);

// get volume of second box

vol = mybox2.volume();

System.out.println("Volume of mybox2 is " + vol);

// get volume of cube

vol = mycube.volume();

System.out.println("Volume of mycube is " + vol);

}

}

OUTPUT

Volume of mybox1 is 3000.0

Volume of mybox2 is -1.0

Volume of mycube is 343.0

# 9. WAP to implement in how many ways a team of 11 player out of 15

import java.util.\*;

public class CriketTeam {

private static long fact(int n)

{

if(n==1)

return 1;

else

return n\*fact(n-1);

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter total no of players : ");

int n = sc.nextInt();

System.out.println("Enter no of players reqd in team : ");

int r = sc.nextInt();

int C = (int) (fact(n) / (fact(n - r) \* fact(r)));

System.out.println("A team of "+r+" players out of "+n+" can be chosen in : "+C+" ways");

}

}

OUTPUT

Enter total no of players : 15

Enter no of players reqd in team : 11

A team of 11 players out of 15 can be chosen in : 1365 ways

# 10. Wap to calculate the distance travelled by a vehicle by s = ut +(1/2\*(a\*(t\*t))

# 11. Program to implment public private & protected mechanism

class base{

private int i;

public int j;

public base() {

i=5;

j=15;

}

protected int add(int a, int b) {

return a+b;

}

public int add() {

return i+j;

}}

class subclass extends base{

public subclass() {

super();

}

public void display() {

System.out.println(add(2,3)+" "+add());

}}

public class q19 {

public static void main(String[] args) {

subclass sub = new subclass();

sub.display();

}

}

OUTPUT

5 20

# 12. WAP for user defined package and interface

//Filename : userpackage/SampleInterface.java

package userpackage;

//user defined package in the folder userpackage

public interface SampleInterface {

public String getName();

public void setName(String name);

}

//Filename : userInterfaceImplement.java

import java.util.Scanner;

import userpackage.SampleInterface;

public class userInterfaceImplement {

public static void main(String arg[]) {

Scanner s = new Scanner(System.in);

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

String name = s.next();

Person person1 = new Person();

person1.setName(name);

System.out.print("Name = " + person1.getName());

}

}

class Person implements SampleInterface {

public String getName() {

return name;

}

public void setName(String n) {

name = "Mr. " + n;

}

private String name;

}

OUTPUT

Enter the name: interface

Name = Mr. interface

# 13 Program to display the records of Books

import java.util.\*;

public class Main {

public static void main(String[] args) {

String[] book = {"java", "toc", "sad"};

int[] price = {100, 200, 202};

Scanner in = new Scanner(System.in);

String ans = "y";

while (ans == null ? "y" == null : ans.equals("y")) {

System.out.println("Enter the book name");

String n = in.nextLine();

int i = 0;

while (i < 3) {

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

if (n == null ? book[i] == null : n.equals(book[i])) {

System.out.println("Price of the book is.." + price[i] + "/=");

break;

} else {

i++;

}

}

System.out.println("Want to check for more book records(y/n)..");

ans = in.nextLine();

}

}

}

OUTPUT:

Enter the book name

java

java

Price of the book is..100/=

Want to check for more book records(y/n)..

y

Enter the book name

toc

java

toc

Price of the book is..200/=

Want to check for more book records(y/n)..

n

# 14. Program to find armstrong number between 100-1000

public class ArmstrongNumber {

public static void main(String[] args) {

System.out.println("Armstrong No. between 100 and 1000 : ");

for (int i = 100; i <= 1000; i++) {

int no = i, sum = 0;

while (no != 0) {

int a = no % 10;

sum = sum + (a \* a \* a);

no = no / 10;

}

if (sum == i) {

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

}}}}

OUTPUT:

Armstrong No. between 100 and 1000 :

153 370 371 407

# 15. Program to implement shift operator

public class ShiftOperator {

public static void main(String[] args) {

java.util.Scanner sc = new java.util.Scanner(System.in);

System.out.println("Enter the dividend");

int n = sc.nextInt();

int quote = n>>1;

System.out.println("Quotient : "+quote);

}

}

OUTPUT:

Enter the dividend

16

Quotient : 8

# 16. Program to show synchronization between threads

//done in question 67

# 17. WAP todemonstrate few types of exception using try and catch blocks

//done in question 42

# 18. WAP to illustrate the use of throw clause

public class THROW\_CLASS {

public void print()

{

System.out.println("WE ARE GOING TO THROW AN EXCEPTION::::");

throw new ArithmeticException(“ARITHMETIC EXCEPTION ”);

}

public static void main(String args[]) {

try

{

new THROW\_CLASS().print( );

}

catch(ArithmeticException ob)

{

System.out.println("ERROR:"+ob.getMessage());

}

}

}

OUTPUT

WE ARE GOING TO THROW AN EXCEPTION

ERROR: ARITHMETIC EXCEPTION

# 19. W.A.P to design a calculator

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class SimpleCalculator {

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

CalculatorFrame frame = new CalculatorFrame();

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

frame.setVisible(true);

}

});}}

class CalculatorFrame extends JFrame {

public CalculatorFrame() {

setTitle("Calculator");

CalculatorPanel panel = new CalculatorPanel();

add(panel);

pack();

}

}

class CalculatorPanel extends JPanel {

public CalculatorPanel() {

setLayout(new BorderLayout());

result = 0;

lastCommand = "=";

start = true;

display = new JButton("0");

display.setEnabled(false);

add(display, BorderLayout.NORTH);

ActionListener insert = new InsertAction();

ActionListener command = new CommandAction();

panel = new JPanel();

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

addButton("7", insert);

addButton("8", insert);

addButton("9", insert);

addButton("/", command);

addButton("4", insert);

addButton("5", insert);

addButton("6", insert);

addButton("\*", command);

addButton("1", insert);

addButton("2", insert);

addButton("3", insert);

addButton("-", command);

addButton("0", insert);

addButton(".", insert);

addButton("=", command);

addButton("+", command);

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 event) {

String input = event.getActionCommand();

if (start) {

display.setText("");

start = false;

}

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

}

}

private class CommandAction implements ActionListener {

public void actionPerformed(ActionEvent event) {

String command = event.getActionCommand();

if (start) {

if (command.equals("-")) {

display.setText(command);

start = false;

} else {

lastCommand = command;

}

} else {

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

lastCommand = command;

start = true;

}}}

public void calculate(double x) {

if (lastCommand.equals("+")) {

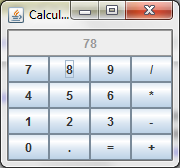
result += x;

} else if (lastCommand.equals("-")) {

result -= x;

} else if (lastCommand.equals("\*")) {

result \*= x;

 } else if (lastCommand.equals("/")) {

result /= x;

} else if (lastCommand.equals("=")) {

result = x;

}

display.setText("" + result);

}

private JButton display;

private JPanel panel;

private double result;

private String lastCommand;

private boolean start;

}

# 20. Program to find power of 2

public class PowerOf2 {

public static void main(String[] args) {

java.util.Scanner sc = new java.util.Scanner(System.in);

System.out.println("Enter the power of 2");

int n = sc.nextInt();

int pow = 1;

for (;n!=0;n--)

{

pow = pow \* 2;

}

System.out.println("Power = "+ pow);

}

}

Output:

Enter the power of 2

4

Power = 16

# 21. WAP to demonstrate the Mouse handling events.

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

/\*

<applet code="Mouse" width=500 height=500>

</applet>

\*/

public class Mouse extends Applet

implements MouseListener,MouseMotionListener

{

int X=0,Y=20;

String msg="MouseEvents";

public void init(){

addMouseListener(this);

addMouseMotionListener(this);

setBackground(Color.black);

setForeground(Color.red);

}

public void mouseEntered(MouseEvent m) {

setBackground(Color.magenta);

showStatus("Mouse Entered");

repaint();

}

public void mouseExited(MouseEvent m){

setBackground(Color.black);

showStatus("Mouse Exited");

repaint();

}

public void mousePressed(MouseEvent m) {

X=10;

Y=20;

msg="SMVDU";

setBackground(Color.green);

repaint();

}

public void mouseReleased(MouseEvent m){

X=10;

Y=20;

msg="Engineering";

setBackground(Color.blue);

repaint();

}

public void mouseMoved(MouseEvent m){

X=m.getX();

Y=m.getY();

msg="College";

setBackground(Color.white);

showStatus("Mouse Moved");

repaint();

}

public void mouseDragged(MouseEvent m) {

msg="CSE";

setBackground(Color.yellow);

showStatus("Mouse Moved"+m.getX()+" "+m.getY());

repaint();

}

public void mouseClicked(MouseEvent m) {

msg="Students";

setBackground(Color.pink);

showStatus("Mouse Clicked");

repaint();

}

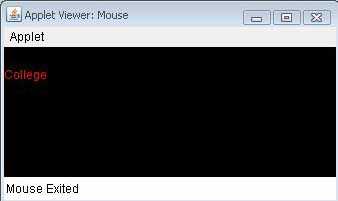
public void paint(Graphics g) {

g.drawString(msg,X,Y);

}

}

Output:



# 22. WAP in awt to demonstrate frame, checkbox, button ,menu of your own choice

import java.awt.\*;

import java.awt.event.\*;

public class AWT extends WindowAdapter {

private Frame f;

private Button b;

private MenuBar bar;

private Menu file,Edit,Help;

private MenuItem New,close;

private Checkbox c1,c2,c3;

public void windowClosing(WindowEvent e)

{

System.exit(0);

}

public void addall()

{

f = new Frame("PROG NO. 22");

f.setSize(450,300);

f.setVisible(true);

f.setLayout(null);

WindowListener listener = new AWT();

f.addWindowListener(listener);

b = new Button("CLOSE");

b.setBounds(200,130,80,40);

f.add(b);

bar = new MenuBar();

file = new Menu("File");

Edit = new Menu("Edit");

Help = new Menu("Help");

New = new MenuItem("New");

close = new MenuItem("Close");

file.add(New);

file.add(close);

bar.add(file);

bar.add(Edit);

bar.add(Help);

// f.add(bar);

f.setMenuBar(bar);

c1 = new Checkbox("YES");

c2 = new Checkbox("NO");

c3 = new Checkbox("CANN'T SAY");

f.add(c1);

 f.add(c2);

f.add(c3);

c1.setBounds(50,80,100,40);

c2.setBounds(150,80,100,40);

c3.setBounds(250,80,100,40);

}

public static void main(String args[]) {

AWT ob = new AWT();

ob.addall();

}

}

OUTPUT:

# 23. Swap two values without using any third variable

import java.util.\*;

public class q1{

public static void main(String args[])

{

Scanner in=new Scanner(System.in);

int a,b;

System.out.println("ENTER THE TWO NOS::");

a=in.nextInt();

b=in.nextInt();

a=a+b;

b=a-b;

a=a-b;

System.out.println("swapped values are a="+a+" "+"b="+b);

} }

# 24. Program to change cms in inch and feet

public class A09 {

public static void main(String[] args) {

java.util.Scanner sc = new java.util.Scanner(System.in);

System.out.println("Enter value in cms");

float cm = sc.nextFloat();

float in = (float) (cm / (2.54));

int ft = (int) (in / 12);

in = in%12;

System.out.println("Feet : "+ft+" Inch : "+in);

}

}

Output :

Enter value in cms

76.2

Feet : 2 Inch : 5.999998

# 25. wap for the string manipulation

import java.util.\*;

public class stringHandling {

public static void main(String args[]) {

String str;

Scanner in = new Scanner(System.in);

System.out.println("ENTER THE STRING");

str = in.next();

System.out.println("STRING LENGTH=" + str.length());

System.out.println("CHANGING THE CASE OF THE STRING=" + str.toUpperCase());

System.out.println("CHARACTER AT 5th INDEX=" + str.charAt(5));

System.out.println("COMPARING WITH hello=" + str.compareTo("hello"));

System.out.println("SUBSTRING=" + str.substring(2, 6));

}

}

Output:

ENTER THE STRING

saurabhkumar

STRING LENGTH=12

CHANGING THE CASE OF THE STRING=SAURABHKUMAR

CHARACTER AT 5th INDEX=b

COMPARING WITH hello=11

SUBSTRING=urab

# 26. Program to implement shift opeartors

import java.util.\*;

public class q6{

public static void main(String args[])

{

Scanner in=new Scanner(System.in);

int n,a;

double l;

System.out.println("ENTER THE NO.::");

a=in.nextInt();

System.out.println("HOW MANY TIMES DO U WANT TO LEFT OR RIGHT SHIFT THE NUMBER::");

n=in.nextInt();

System.out.println("THROUGH LEFT SHIFT THE NO.="+(a<<n));

l=a>>n;

System.out.println("THROUGH RIGHT SHIFT THE NO.="+l);

}

}

OUTPUT

ENTER THE NO.::59

HOW MANY TIME DO U WANT TO LEFT OR RIGHT SHIFT THE NUMBER::3

THROUGH LEFT SHIFT THE NO.=472

THROUGH RIGHT SHIFT THE NO.=7.0

# 27. Program to search an element in an array

import java.util.\*;

public class Search {

public static void main(String args[]) {

System.out.print("ENTER THE TOTAL NO. OF ELEMENT IN ARRAY = ");

Scanner ob = new Scanner(System.in);

int flag = 0,i;

int n = ob.nextInt();

int arr[] = new int[n];

for( i = 0 ; i < n ; i ++)

{

System.out.print("ENTER THE "+(i+1)+" ELEMENT = ");

arr[i] = ob.nextInt();

}

System.out.print("ENTER THE NUMBER U WANT TO SEARCH = ");

int s = ob.nextInt();

for( i = 0 ; i < n ; i ++)

{

if(arr[i] == s )

{

flag = 1;

break;

}

}

if( flag == 1)

{

System.out.println("ELEMENT FOUND IN ARRAY AT "+(i+1)+" POSITION");

}

else

System.out.println("ELEMENT NOT FOUND IN ARRAY:::");

}

}

OUTPUT::

ENTER THE TOTAL NO. OF ELEMENT IN ARRAY = 4

ENTER THE 1 ELEMENT = 4

ENTER THE 2 ELEMENT = 3

ENTER THE 3 ELEMENT = 10

ENTER THE 4 ELEMENT = 2

ENTER THE NUMBER U WANT TO SEARCH = 10

ELEMENT FOUND IN ARRAY AT 3 POSITION

# 28. Program to solve linear & quadratic equation

import java.util.\*;

public class q13

{

public static void main(String args[])

{

Scanner in=new Scanner(System.in);

double a,b,a1,b1,c1,x1,x2,x3;

double t1,t2;

System.out.println("FOR LINEAR EQUATION::INPUT VALUES OF a & b in ax+b=0");

a=in.nextDouble();

b=in.nextDouble();

x1=(-b)/a;

System.out.println("SOLUTION OF LINEAR EQN="+x1);

System.out.println("FOR QUADRATIC EQN::INPUT VALUES OF a,b&c in ax^2+bx+c=0");

a1=in.nextDouble();

b1=in.nextDouble();

c1=in.nextDouble();

t1=(b1\*b1-4\*a1\*c1);

if(t1>0)

{

t2=Math.sqrt(t1);

x2=(-b1+t2)/(2\*a1);

x3=(-b1-t2)/(2\*a1);

System.out.println("SOLUTION OF QUADRATIC EQUATION="+x2+" & "+x3);

}

else

{

t1=-t1;

t2=Math.sqrt(t1);

System.out.println("SOLUTION OF QUADRATIC EQUATION x1="+(-b1/(2\*a1))+"+i"+(t2/(2\*a1)));

System.out.println("SOLUTION OF QUADRATIC EQUATION x2="+(-b1/(2\*a1))+"-i"+(t2/(2\*a1)));

}

}

}

OUTPUT

FOR LINEAR EQUATION::INPUT VALUES OF a & b in ax+b=0::2 3

SOLUTION OF LINEAR EQN=-1.5

FOR QUADRATIC EQN::INPUT VALUES OF a,b&c in ax^2+bx+c=0::1 4 6

SOLUTION OF QUADRATIC EQUATION x1=-2.0+i1.4142135623730951

SOLUTION OF QUADRATIC EQUATION x2=-2.0-i1.4142135623730951

# 29. WAP using default argument constructor

//done in question 3

# 30 . WAP to compute whether the given no. is prime or not

import java.util.\*;

public class q15

{

public static void main(String args[])

{

Scanner in=new Scanner(System.in);

int n,f=1;

System.out.println("ENTER THE NO.::");

n=in.nextInt();

for(int i=2;i<=n/2;i++)

{

if(n%i==0)

{

f=0;

break;

}

}

if(f==0)

System.out.println("NUMBER IS NOT PRIME");

else

System.out.println("NUMBER IS PRIME");

}

}

OUTPUT:

ENTER THE NUMBER::17

NUMBER IS PRIME

# 31. Program to find the power of 2 without arithematic operation

import java.util.Scanner;

public class power {

public static void main(String args[]) {

System.out.print("Enter the Number = ");

Scanner ob = new Scanner(System.in);

int i = ob.nextInt();

int n = 1;

while( i!= 2)

{

i = i>>1;

n++;

}

System.out.println("Power of 2 = "+n);

}

}

OUTPUT:

ENTER THE NUMBER = 32

POWER OF 2 = 5

# 32. Program to find permutation p(n,r)=n!/(n-r)!

import java.util.\*;

public class q21 {

private static int fact(int n)

{

if(n==1)

return 1;

else

return n\*fact(n-1);

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("For Permutation nPr Enter the value of n & r : ");

int n = sc.nextInt();

int r = sc.nextInt();

int P = fact(n)/fact(n-r);

System.out.println("Permutation "+n+"P"+r+" : "+P);

}

}

OUTPUT

For Permutation nPr Enter the value of n & r ::7 3

Permutation 7P3 : 210

# 33. Program to find the transpose of a Matrix

import java.util.\*;

public class Main {

public static void main(String[] args) {

int i, j, n, m, temp;

Scanner in = new Scanner(System.in);

System.out.println("Enter no. of rows");

n = in.nextInt();

System.out.println("Enter no. of columns");

m = in.nextInt();

int[][] mat1 = new int[n][m];

int[][] mat2 = new int[n][m];

for (i = 0; i < n; i++) {

for (j = 0; j < m; j++) {

System.out.println("Enter (mat" + i + "," + j + ")");

mat1[i][j] = in.nextInt();

}

}

System.out.println("original array is");

for (i = 0; i < n; i++) {

for (j = 0; j < m; j++) {

System.out.print(mat1[i][j] + " ");

}

System.out.println();

}

for (i = 0; i < n; i++) {

for (j = 0; j < m; j++) {

mat2[i][j] = mat1[j][i];

}

}

System.out.println("transposed array is");

for (i = 0; i < n; i++) {

for (j = 0; j < m; j++) {

System.out.print(mat2[i][j] + " ");

}

System.out.println();

}

}

}

OUTPUT:

Enter no. of rows

2

Enter no. of columns

2

Enter (mat0,0)

1

Enter (mat0,1)

2

Enter (mat1,0)

3

Enter (mat1,1)

4

original array is

1 2

3 4

transposed array is

1 3

2 4

# 34. Program to display the faculy Records

import java.util.\*;

class teachers {

String[] name = {"neeraj nehra", "sanjay sharma", "s.b kotwal", "sadhna tiwari", "ashish suri"};

String[] dept = {"CSE", "CSE", "ENC", "MATHEMATICS", "ENC"};

}

public class Main extends teachers {

public static void main(String[] args) {

int i;

String ans = "y";

teachers obj = new teachers();

Scanner in = new Scanner(System.in);

while (ans == null ? "y" == null : ans.equals("y")) {

System.out.println("Enter the faculty name..");

String n = in.nextLine();

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

if (n == null ? obj.name[i] == null : n.equals(obj.name[i])){

System.out.println("Department is.." + obj.dept[i]);

break;

}}

if (i == 5) {

System.out.println("Record not found");

}

System.out.println("Want to check for more records(y/n)");

ans = in.nextLine();

}}}

OUTPUT:

Enter the faculty name..

neeraj nehra

Department is..CSE

Want to check for more records(y/n)

y

Enter the faculty name..

sadhna

Record not found

Want to check for more records(y/n)

n

# 35. Program for constructor overloading

class Box {

double width;

double height;

double depth;

// constructor used when all dimensions specified

Box(double w, double h, double d) {

width = w;

height = h;

depth = d;

}

// constructor used when no dimensions specified

Box() {

width = -1; // use -1 to indicate

height = -1; // an uninitialized

depth = -1; // box

}

// constructor used when cube is created

Box(double len) {

width = height = depth = len;

}

// compute and return volume

double volume() {

return width \* height \* depth;

}

}

class q18 {

public static void main(String args[]) {

// create boxes using the various constructors\

Box mybox1 = new Box(10, 20, 15);

Box mybox2 = new Box();

Box mycube = new Box(7);

double vol;

// get volume of first box

vol = mybox1.volume();

System.out.println("Volume of mybox1 is " + vol);

// get volume of second box

vol = mybox2.volume();

System.out.println("Volume of mybox2 is " + vol);

// get volume of cube

vol = mycube.volume();

System.out.println("Volume of mycube is " + vol);

}

}

OUTPUT

Volume of mybox1 is 3000.0

Volume of mybox2 is -1.0

Volume of mycube is 343.0

# 36. Program to demonstrate linked list

import java.util.\*;

class q28

{

public static void main(String args[])

{

// create a linked list

LinkedList ll = new LinkedList();

// add elements to the linked list

ll.add("F");

ll.add("B");

ll.add("D");

ll.add("E");

ll.add("C");

ll.addLast("Z");

ll.addFirst("A");

ll.add(1, "A2");

System.out.println("Original contents of ll: " + ll);

// remove elements from the linked list

ll.remove("F");

ll.remove(2);

System.out.println("Contents of ll after deletion: "+ ll);

// remove first and last elements

ll.removeFirst();

ll.removeLast();

System.out.println("ll after deleting first and last: "+ ll);

// get and set a value

Object val = ll.get(2);

ll.set(2, (String) val + " Changed");

System.out.println("ll after change: " + ll);

}

}

OUTPUT

Original contents of ll: [A, A2, F, B, D, E, C, Z]

Contents of ll after deletion: [A, A2, D, E, C, Z]

ll after deleting first and last: [A2, D, E, C]

ll after change: [A2, D, E Changed, C]

# 37. Solving a linear equation

//Done in question 28

# 38. Program to implement String Tokenizer

import java.util.StringTokenizer;

public class StringTokenizerClass {

public static void main(String[] args) {

String s1 ="Java Programming in Netbeans ";

StringTokenizer st = new StringTokenizer(s1);

while(st.hasMoreTokens())

System.out.println(st.nextToken());

String s2 ="SMVDU:COE:CSE;Vth SEM;BEST";

StringTokenizer st1 = new StringTokenizer(s2,";:");

while(st1.hasMoreTokens())

System.out.println(st1.nextToken());

}}

OUTPUT

Java

Programming

in

Netbeans

SMVDU

COE

CSE

Vth SEM

BEST

# 39. Program to implement final, static and abstract modifier

abstract class shape

{

static float rad = 5;

final float peri(){

return (float) (6.28 \* rad);

}

abstract double area();

}

class circle extends shape

{

double area() {

return 3.14\*rad\*rad;

}

}

public class A17 {

public static void main(String[] args) {

circle c = new circle();

System.out.println("Area : "+c.area()+" Perimeter :"+c.peri());

}

}

Output :

Area : 78.5 Perimeter : 31.4

# 40. Program for implementing Fibonacci series

import java.util.\*;

public class q8

{

public static void main(String args[])

{

Scanner in=new Scanner(System.in);

int n,i=1;

int a=0,b=1,c;

System.out.println("ENTER THE LIMIT");

n=in.nextInt();

System.out.print("FIBONACCI SERIES::");

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

do

{

c=a+b;

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

a=b;

b=c;

i++;

}while(i<=(n-2));

}

}

OUTPUT

ENTER THE LIMIT:: 8

FIBONACCI SERIES::0 1 1 2 3 5 8 13

# 41. Write a program of ur own which implements interface

interface one {

void print1();

void print2();

}

interface two {

void print3();

}

class face implements two {

public void print1() {

System.out.println("ONE::1");

}

public void print2() {

System.out.println("TWO::2");

}

public void print3() {

System.out.println("THREE::3");

}

}

public class q23 {

public static void main(String args[]) {

face f = new face();

f.print1();

f.print2();

f.print3();

}

}

OUTPUT

ONE::1

TWO::2

THREE::3

# 42. Write a program to show at least 8 exceptions.

import java.net.\*;

import java.io.\*;

class abc {

int a = 10;

void show() {

System.out.println(a);

}

}

public class ExceptionTest {

public static void main(String args[]) {

try {//excep 1

URL u = new URL("abc://www.google.com");

} catch (MalformedURLException e) {

e.printStackTrace();

}

try{//excep2

InetAddress in = InetAddress.getByName("www.gaagle.com");

System.out.println("ip of google::" + in);

} catch (UnknownHostException e) {

e.printStackTrace();

}

try {//excep3

int a = 10, b = 0, c;

c = a / b;

} catch (ArithmeticException e) {

e.printStackTrace();

}

Try{ //excep4

int a[] = new int[5];

System.out.println(a[5]);

} catch (ArrayIndexOutOfBoundsException e) {

e.printStackTrace();

}

try { //excep5

FileInputStream in = new FileInputStream("hello.txt");

} catch (FileNotFoundException e) {

e.printStackTrace();

}

try { //excep6

Socket s = new Socket("localhost", 4567);

} catch (Exception e) {

e.printStackTrace();

}

try{ //excep7

int a[] = new int[-7];

} catch (NegativeArraySizeException e) {

e.printStackTrace();

}

try { //excep8

String str = "hello";

System.out.println(str.charAt(7));

} catch (StringIndexOutOfBoundsException e) {

e.printStackTrace();

}

}

}

OUTPUT

java.net.MalformedURLException: unknown protocol: abc

java.net.UnknownHostException: www.gaagle.com

java.lang.ArithmeticException: / by zero

java.lang.ArrayIndexOutOfBoundsException: 5

java.io.FileNotFoundException: hello.txt (No such file or directory)

java.net.ConnectException: Connection refused

java.lang.NegativeArraySizeException

java.lang.StringIndexOutOfBoundsException: String index out of range: 7

# 43. Program to compute division of 2 without using any mathmatical operator

import java.util.\*;

public class q3

{

public static void main(String args[])

{

Scanner in=new Scanner(System.in);

int n;

double d;

System.out.println("ENTER THE NUMBER WHICH U WANT TO DIVIDE");

n=in.nextInt();

d=n>>1;

System.out.println("DIVISION="+d);

}

}

OUTPUT

ENTER THE NUMBER WHICH U WANT TO DIVIDE::23

DIVISION=11.0

# 44. Program to implement Runtime polymorphism

class A {

public void display() {

System.out.println("In A class");

}

}

class sub1 extends A {

public void display() {

System.out.println("In Subclass1 class");

}

}

class sub2 extends A {

public void display() {

System.out.println("In Subclass2 class");

}

}

public class q20 {

public static void main(String[] args) {

A b;

A b1 = new A();

sub1 s1 = new sub1();

sub2 s2 = new sub2();

b = b1;

b.display();

b = s1;

b.display();

b = s2;

b.display();

}

}OUTPUT

In A class

In Subclass1 class

In Subclass2 class

# 45. Program for a frog to reach the top of wall.

public class A12 {

public static void main(String[] args) {

java.util.Scanner sc = new java.util.Scanner(System.in);

System.out.println("Enter the height of wall ");

int a = sc.nextInt();

System.out.println("Enter Climbing distance ");

int b = sc.nextInt();

System.out.println("Enter Skiding distance ");

int c = sc.nextInt();

int net = 0,hr = 0;

while(net<(a-b)){ hr++; net = net + (b-c);}

hr++;

System.out.println("Time taken to reach top : "+hr+" hrs.");

}}

Output :

Enter the height of wall

30

Enter Climbing distance

3

Enter Skiding distance

2

Time taken to reach top : 28 hrs.

# 46. WAP to find the sum of Digits

public class sum {

public static void main(String[] args) {

java.util.Scanner sc = new java.util.Scanner(System.in);

System.out.println("Enter the Number");

int n = sc.nextInt();

int sum = 11;

while(sum > 10){

sum = 0;

while(n!=0){

sum = sum + (n%10);

n = n/10;

}

n=sum;

}

System.out.println("Sum of digits : "+sum);

}}

Output:

Enter the Number

12345

Sum of digits : 6

# 47. Program to find the combination 15C11

// Same as question 9

# 48. Program to implement public, private and protected mechanism

class base {

private int i;

public int j;

public base() {

i = 5;

j = 15;

}

protected int add(int a, int b) {

return a + b;

}

public int add() {

return i + j;

}

}

class subclass extends base {

public subclass() {

super();

}

public void display() {

System.out.println(add(2, 3) + " " + add());

}

}

public class A19 {

public static void main(String[] args) {

subclass sub = new subclass();

sub.display();

}

}

# 49. Program to check whether the given character is a vowel or not

import java.util.\*;

public class q7 {

public static void main(String args[]) {

Scanner in=new Scanner(System.in);

String ch="";

int f=0;

System.out.println("ENTER THE CHARACTER::");

ch=in.next();

char c=ch.charAt(0);

switch(c){ case 'a':

case 'A':

case 'e':

case 'E':

case 'i':

case 'I':

case 'o':

case 'O':

case 'u':

case 'U':

f=1; break;

}

if(f==1)

System.out.println(ch+" IS A VOWEL CHARACTER");

else

System.out.println(ch+" IS NOT A VOWEL CHARACTER");

}

}

OUTPUT

ENTER THE CHARACTER::u

u IS A VOWEL CHARACTER

# 50. W.A.P to create an applet with button

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

public class appletdemo extends Applet implements ActionListener {

String msg = "";

Button yes, no, maybe;

public void init() {

yes = new Button("Yes");

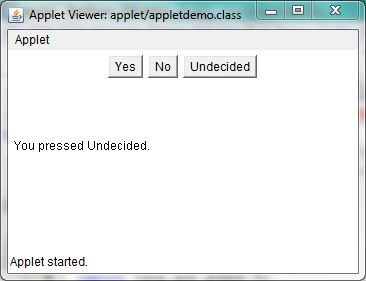
no = new Button("No");

maybe = new Button("Undecided");

add(yes);

add(no);

add(maybe);

yes.addActionListener(this);

no.addActionListener(this);

maybe.addActionListener(this);

}

public void actionPerformed(ActionEvent ae)

{

String str = ae.getActionCommand();

if(str.equals("Yes")) {

msg = "You pressed Yes.";

}

else if(str.equals("No")) {

msg = "You pressed No.";

}

else {

msg = "You pressed Undecided.";

}

repaint();

}

public void paint(Graphics g) {

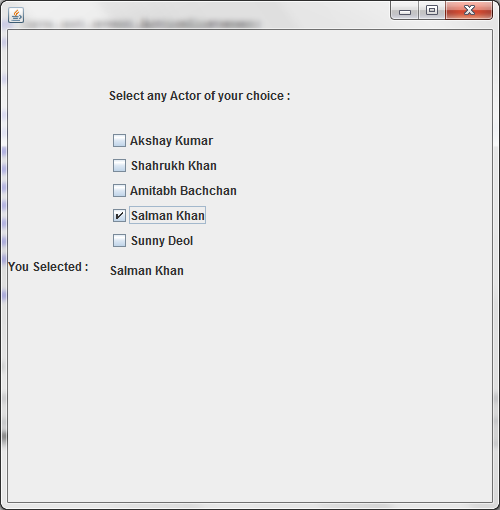
g.drawString(msg, 6, 100);

}}

# 51. Program to display actor's name in checkboxes

import java.awt.event.ActionListener;

import javax.swing.\*;



public class A33A extends JFrame {

private ButtonGroup buttonGroup1;

private static JLabel disp;

private static JCheckBox jCheckBox1;

private static JCheckBox jCheckBox2;

private static JCheckBox jCheckBox3;

private static JCheckBox jCheckBox4;

private static JCheckBox jCheckBox5;

private static JLabel jLabel1;

private static JLabel jLabel2;

public A33A() {

init();

this.setBounds(0, 0, 500, 510);

this.setLocationRelativeTo(null);

}

private void init() {

buttonGroup1 = new ButtonGroup();

disp = new JLabel();

jCheckBox1 = new JCheckBox("Amitabh Bachchan");

jCheckBox2 = new JCheckBox("Akshay Kumar");

jCheckBox3 = new JCheckBox("Shahrukh Khan");

jCheckBox4 = new JCheckBox("Salman Khan");

jCheckBox5 = new JCheckBox("Sunny Deol");

jLabel1 = new JLabel("Select any Actor of your choice : ");

jLabel2 = new JLabel("You Selected :");

ActionListener al = new ActionListener() {

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

if (jCheckBox1.isSelected()) {

disp.setText("Amitabh Bachchan");

} else if (jCheckBox2.isSelected()) {

disp.setText("Akshay Kumar");

} else if (jCheckBox3.isSelected()) {

disp.setText("Shahrukh Khan");

} else if (jCheckBox4.isSelected()) {

disp.setText("Salman Khan");

} else if (jCheckBox5.isSelected()) {

disp.setText("Sunny Deol");

}

}

};

buttonGroup1.add(jCheckBox1);

buttonGroup1.add(jCheckBox2);

buttonGroup1.add(jCheckBox3);

buttonGroup1.add(jCheckBox4);

buttonGroup1.add(jCheckBox5);

jCheckBox5.addActionListener(al);

jCheckBox4.addActionListener(al);

jCheckBox3.addActionListener(al);

jCheckBox2.addActionListener(al);

jCheckBox1.addActionListener(al);

}

public static void main(String[] args) {

A33A a = new A33A();

a.setVisible(true);

a.add(disp).setBounds(101, 225, 200, 30);

a.add(jCheckBox1).setBounds(101, 150, 200, 20);

a.add(jCheckBox2).setBounds(101, 100, 200, 20);

a.add(jCheckBox3).setBounds(101, 125, 200, 20);

a.add(jCheckBox4).setBounds(101, 175, 200, 20);

a.add(jCheckBox5).setBounds(101, 200, 200, 20);

a.add(jLabel1).setBounds(100, 50, 300, 30);

a.add(jLabel2).setBounds(5, 225, 100, 30);

}}

# 52. Program to demonstrate the User-Defined Exceptions.

class MyException extends Exception {

public MyException() { super(); }

public MyException(String s) { super(s); }

}

class MyOtherException extends Exception {

public MyOtherException() { super(); }

public MyOtherException(String s) { super(s); }

}

class MySubException extends MyException {

public MySubException() { super(); }

public MySubException(String s) { super(s); }

}

public class throwtest {

public static void main(String argv[]) {

int i;

try {

i = Integer.parseInt(argv[0]);

}

catch (ArrayIndexOutOfBoundsException e) { // argv is empty

System.out.println("Must specify an argument");

return;

}

catch (NumberFormatException e) { // argv[0] isn't an integer

System.out.println("Must specify an integer argument.");

return;

}

// Now, pass that integer to method a().

a(i);

}

public static void a(int i) {

try {

b(i);

}

catch (MyException e) { // Point 1.

// Here we handle MyException and

// its subclass MyOtherException

if (e instanceof MySubException)

System.out.print("MySubException: ");

else

System.out.print("MyException: ");

System.out.println(e.getMessage());

System.out.println("Handled at point 1");

}

}

public static void b(int i) throws MyException {

int result;

try {

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

result = c(i);

System.out.print(" c(i) = " + result);

}

catch (MyOtherException e) { // Point 2

// Handle MyOtherException exceptions:

System.out.println("MyOtherException: " + e.getMessage());

System.out.println("Handled at point 2");

}

finally {

// Terminate the output we printed above with a newline.

System.out.print("\n");

}

}

public static int c(int i) throws MyException, MyOtherException {

switch (i) {

case 0: // processing resumes at point 1 above

throw new MyException("input too low");

case 1: // processing resumes at point 1 above

throw new MySubException("input still too low");

case 99:// processing resumes at point 2 above

throw new MyOtherException("input too high");

default:

return i\*i;

}

}

}

# 53. Program to implement bitwise operators

import java.util.\*;

public class q5{

public static void main(String args[])

{

Scanner in=new Scanner(System.in);

System.out.println("ENTER TWO NOS::");

int a,b;

a=in.nextInt();

b=in.nextInt();

System.out.println("BITWISE OR="+(a|b));

System.out.println("BITWISE AND="+(a&b));

System.out.println("BITWISE NOT OF a="+(~a));

System.out.println("BITWISE NOT OF b="+(~b));

System.out.println("BITWISE XOR="+(a^b));

}}

OUTPUT

ENTER TWO NOS::20 30

BITWISE OR=30

BITWISE AND=20

BITWISE NOT OF a=-21

BITWISE NOT OF b=-31

BITWISE XOR=10

# 54. WAP to create an applet to receive text from textfield and display on applet

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

public class appletdemo2 extends Applet implements ActionListener {

TextField name;

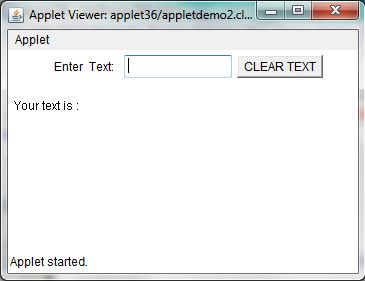
public void init() {

Label namep = new Label("Enter Text: ", Label.RIGHT);

Button button = new Button("CLEAR TEXT");

name = new TextField(12);

add(namep);

add(name);

add(button,BorderLayout.SOUTH);

name.addActionListener(this);

button.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

name.setText(null);

repaint();

}});}

public void actionPerformed(ActionEvent ae) {

repaint();

}

public void paint(Graphics g) {

g.drawString("Your text is : " + name.getText(), 6, 60);

}}

# 55. Program to create Menu in a Frame.

// done in question 39

# 56. Program to implement Vector class

import java.util.\*;

class VectorClass

{

public static void main(String args[])

{

// initial size is 3, increment is 2

Vector v = new Vector(3, 2);

System.out.println("Initial size: " + v.size());

System.out.println("Initial capacity: " +

v.capacity());

v.addElement(new Integer(1));

v.addElement(new Integer(2));

v.addElement(new Integer(3));

v.addElement(new Integer(4));

System.out.println("Capacity after four additions: " +

v.capacity());

v.addElement(new Double(5.45));

System.out.println("Current capacity: " +

v.capacity());

v.addElement(new Double(6.08));

v.addElement(new Integer(7));

System.out.println("Current capacity: " +

v.capacity());

v.addElement(new Float(9.4));

v.addElement(new Integer(10));

System.out.println("Current capacity: " +

v.capacity());

v.addElement(new Integer(11));

v.addElement(new Integer(12));

System.out.println("First element: " +(Integer)v.firstElement());

System.out.println("Last element: " +(Integer)v.lastElement());

if(v.contains(new Integer(3)))

System.out.println("Vector contains 3.");

// enumerate the elements in the vector.

Enumeration vEnum = v.elements();

System.out.println("\nElements in vector:");

while(vEnum.hasMoreElements())

System.out.print(vEnum.nextElement() + " ");

System.out.println();

}

}

OUTPUT

Initial size: 0

Initial capacity: 3

Capacity after four additions: 5

Current capacity:5

Current capacity: 7

Current capacity: 9

First element: 1

Last element: 12

Vector contains 3.

Elements in vector:

1 2 3 4 5.45 6.08 7 9.4 10 11 12

# 58. Program to implement type casting

import java.util.\*;

public class q9{

public static void main(String args[]) {

int a;

double b;

Scanner in=new Scanner(System.in);

System.out.println("INPUT AN INTEGER VALUE::");

a=in.nextInt();

System.out.println("INPUT A DOUBLE VALUE::");

b=in.nextDouble();

System.out.println(a+" in byte="+(byte)a);

System.out.println(b+" in int="+(int)b);

System.out.println(b+" in byte="+(byte)b);

}

}

OUTPUT

INPUT AN INTEGER VALUE::567

INPUT A DOUBLE VALUE::4567.68677

567 in byte=55

4567.68677 in int=4567

4567.68677 in byte=-41

# 59. Program to display car in list

import javax.swing.\*;

import javax.swing.event.ListSelectionEvent;

import javax.swing.event.ListSelectionListener;

public class CarList extends JFrame {

private JList jList1;

private JLabel jLabel2;

private JLabel disp;

private JScrollPane js;

public CarList() {

initComponents();

setBounds(0, 0, 500, 500);

setLocationRelativeTo(null);

}

private void initComponents() {

jList1 = new JList();

add(jList1);

jLabel2 = new JLabel("You Selected : ");

add(jLabel2).setBounds(100, 150, 100, 25);

disp = new JLabel();

add(disp).setBounds(185, 150, 100, 25);

js = new JScrollPane();

add(js);

js.setViewportView(jList1);

jList1.setModel(new AbstractListModel() {

String[] strings = {"Mercedes Benz", "Maruti Alto",

"Hyundai i10", "Chevrolet Spark", "Tata Safari"};

public int getSize() {

return strings.length;

}

public Object getElementAt(int i) {

return strings[i];

}});

jList1.addListSelectionListener(new ListSelectionListener() {

public void valueChanged(ListSelectionEvent evt) {

jList1ValueChanged(evt);

}});

}

private void jList1ValueChanged(ListSelectionEvent evt) {

disp.setText(jList1.getSelectedValue().toString());

}

public static void main(String[] a) {

new CarList().setVisible(true);

}}

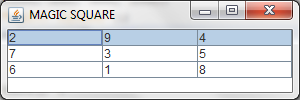
# 60. Program to show magic square in table

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

public class magic extends JFrame {

private JTable jt;

 private JScrollPane jsp;

magic() {

init();

setBounds(100, 100, 300, 100);

setTitle("MAGIC SQUARE");

setLocationRelativeTo(null);

}

private void init() {

jt = new JTable();

jsp = new JScrollPane();

jsp.setViewportView(jt);

jt.setModel(new DefaultTableModel(

new Object[][]{

{2, 9, 4},

{7, 3, 5},

{6, 1, 8}

},

new String[]{

"MAGIC", "SQUARE", "TABLE"

}));

add(jt).setBounds(100, 100, 300, 100);

}

public static void main(String[] a) {

new magic().setVisible(true);

}

}

# 61. Program to create a thread by implementing runnable interface

class NewThread implements Runnable{

Thread t;

public NewThread(){

t = new Thread(this,"Demo Thread");

t.start();

}

public void run() {

System.out.println("Thread Running");

}}

public class A66 {

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

NewThread nt = new NewThread();

nt.t.join();

}

}

OUTPUT

Thread Running

# 62. Program to show synchronization between threads

class Q{

int n;

boolean valueSet = false;

synchronized int get() {

if(!valueSet)

try{

wait();

} catch(InterruptedException e) {

System.out.println("InterruptedException caught");

}

System.out.println("Got: " + n);

valueSet = false;

notify();

return n;

}

synchronized void put(int n) {

if(valueSet)

try {

wait();

} catch(InterruptedException e) {

System.out.println("InterruptedException caught");

}

this.n = n;

valueSet = true;

System.out.println("Put: " + n);

notify();

}}

class Producer implements Runnable {

Q q;

Producer(Q q) {

this.q = q;

new Thread(this, "Producer").start();

}

public void run() {

int i = 0;

while(true) {

q.put(i++);

}}}

class Consumer implements Runnable {

Q q;

Consumer(Q q) {

this.q = q;

new Thread(this, "Consumer").start();

}

public void run() {

while(true) {

q.get();

}}}

class q31 {

public static void main(String args[]) {

Q q = new Q();

new Producer(q);

new Consumer(q);

System.out.println("Press Control-C to stop.");

}}