**JAVA RECORD PROGRAMS**

1. **WAP to find factorial of a given using any loop.**

import java.util.\*;

class factorial{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

System.out.println(“enter a number:”);

Int number = sc.nextInt();

int i,fact=1;

for(i=1;i<=number;i++){

fact=fact\*i;

}

System.out.println("Factorial of "+number+" is: "+fact);

}

}

1. **WAP to find biggest of 3 numbers using nested-if.**

import java.util.\*;

public class JavaExample{

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("enter any three numbers:");

int num1 = sc.nextInt();

int num2 = sc.nextInt();

int num3 = sc.nextInt();

if(num1 >= num2) {

if(num1 >= num3)

System.out.println(num1+" is the largest Number");

else

System.out.println(num3+" is the largest Number");

}

else {

if(num2 >= num3)

System.out.println(num2+" is the largest Number");

else

System.out.println(num3+" is the largest Number");

}

}

}

1. **WAP to find biggest of 3 numbers using if-ladder.**

import java.util.\*;

public class biggest\_if\_ladder

{

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("enter any three numbers:");

int num1 = sc.nextInt();

int num2 = sc.nextInt();

int num3 = sc.nextInt();

if( num1 >= num2 && num1 >= num3)

System.out.println(num1+" is the largest Number");

else if (num2 >= num1 && num2 >= num3)

System.out.println(num2+" is the largest Number");

else

System.out.println(num3+" is the largest Number");

}

}

1. **WAP to display VIBGYOR using switch-case.**

import java.util.Scanner;

public class vibgyor

{

public static void main(String args[])

{

char ch;

System.out.println("enter color code in VIBGYOR ");

Scanner sc= new Scanner(System.in);

ch = sc.next().charAt(0);

switch(ch)

{

case 'v':

System.out.println("violet");

break;

case 'i':

System.out.println("indigo");

break;

case 'b':

System.out.println("blue");

break;

case 'g':

System.out.println("green");

break;

case 'y':

System.out.println("yellow");

break;

case 'o':

System.out.println("orange");

break;

case 'r':

System.out.println("red");

break;

default:

System.out.println("labled color code");

break;

}

}

}

1. **WAP to implement class using following information:**

**Student roll no, Student name, Student course, any 5 subjects, and calculate total & average.**

import java.io.\*;

import java.util.\*;

class student{

Scanner sc = new Scanner (System.in);

int rno, total=0, sub;

float avg=0;

String name, course;

int marks[];

student()

{

System.out.println("Enter Roll no: ");

rno=sc.nextInt();

System.out.println("Enter Student name: ");

name = sc.next();

System.out.println("Enter Student course: ");

course = sc.next();

getmarks();

}

public void getmarks()

{

marks= new int[5];

System.out.println("Enter marks of WT: ");

marks[0]=sc.nextInt();

System.out.println("Enter marks of JAVA: ");

marks[1]=sc.nextInt();

System.out.println("Enter marks of SE: ");

marks[2]=sc.nextInt();

System.out.println("Enter marks od PSY: ");

marks[3]=sc.nextInt();

System.out.println("Enter marks of CD: ");

marks[4]=sc.nextInt();

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

{

total+=marks[i];

}

avg=total/5;

System.out.println("Total marks of the student "+name+": "+total);

System.out.println("Average="+avg);

}

}

public class studentinfo{

public static void main(String args[])

{

student s=new student();

}

}

1. **WAP to generate a Fibonacci sequence using any loop.**

import java.io.\*;

import java.util.\*;

public class FibonacciExample {

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

System.out.println("enter number:");

int maxNumber = sc.nextInt();

int previousNumber = 0;

int nextNumber = 1;

System.out.print("Fibonacci Series of "+maxNumber+" numbers:");

for (int i = 1; i <= maxNumber; ++i)

{

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

int sum = previousNumber + nextNumber;

previousNumber = nextNumber;

nextNumber = sum;

}

}

}

1. **WAP to add 2 matrices.**

import java.io.\*;

import java.util.\*;

class addmat

{

public static void main(String args[])

{

int row, col,i,j;

Scanner in = new Scanner(System.in);

System.out.println("Enter the number of rows");

row = in.nextInt();

System.out.println("Enter the number columns");

col = in.nextInt();

int mat1[][] = new int[row][col];

int mat2[][] = new int[row][col];

int res[][] = new int[row][col];

System.out.println("Enter the elements of matrix1");

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

{

for ( j= 0 ; j < col ;j++ )

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

System.out.println();

}

System.out.println("Enter the elements of matrix2");

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

{

for ( j= 0 ; j < col ;j++ )

mat2[i][j] = in.nextInt();

System.out.println();

}

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

for ( j= 0 ; j < col ;j++ )

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

System.out.println("Sum of matrices:-");

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

{

for ( j= 0 ; j < col ;j++ )

System.out.print(res[i][j]+"\t");

System.out.println();

}

}

}

1. **WAP to multiply 2 matrices.**

import java.io.\*;

import java.util.\*;

class mulmat

{

public static void main(String args[])

{

int r1, r2,c1,c2,i,j,k,sum;

Scanner in = new Scanner(System.in);

System.out.println("Enter the number of rows of matrix1");

r1 = in.nextInt();

System.out.println("Enter the number columns of matrix 1");

c1 = in.nextInt();

System.out.println("Enter the number of rows of matrix2");

r2 = in.nextInt();

System.out.println("Enter the number of columns of matrix 2");

c2 = in.nextInt();

if(c1==r2)

{

int mat1[][] = new int[r1][c1];

int mat2[][] = new int[r2][c2];

int res[][] = new int[r1][c2];

System.out.println("Enter the elements of matrix1");

for ( i= 0 ; i < r1 ; i++ )

{

for ( j= 0 ; j < c1 ;j++ )

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

}

System.out.println("Enter the elements of matrix2");

for ( i= 0 ; i < r2 ; i++ )

{

for ( j= 0 ; j < c2 ;j++ )

mat2[i][j] = in.nextInt();

}

System.out.println("\n\noutput matrix:-");

for ( i= 0 ; i < r1 ; i++ )

for ( j= 0 ; j <c2;j++)

{

sum=0;

for ( k= 0 ; k <r2;k++ )

{

sum +=mat1[i][k]\*mat2[k][j] ;

}

res[i][j]=sum;

}

for ( i= 0 ; i < r1; i++ )

{

for ( j=0 ; j < c2;j++ )

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

System.out.println();

}

}

else

System.out.print("multipication does not exist ");

}

}

1. **WAP to check if the given string is palindrome or not.**

import java.io.\*;

import java.util.\*;

class palindrome

{

public static void main(String args[])

{

String a, b = "";

Scanner s = new Scanner(System.in);

System.out.print("Enter the string you want to check:");

a = s.nextLine();

int n = a.length();

for(int i = n - 1; i >= 0; i--)

{

b = b + a.charAt(i);

}

if(a.equalsIgnoreCase(b))

{

System.out.println("The string is palindrome.");

}

else

{

System.out.println("The string is not palindrome.");

}

}

}

1. **WAP to sort the given list of names in ascending order.**

import java.util.\*;

public class Alphabetical\_Order

{

public static void main(String[] args)

{

int n;

String temp;

Scanner s = new Scanner(System.in);

System.out.print("Enter number of names you want to enter:");

n = s.nextInt();

String names[] = new String[n];

Scanner s1 = new Scanner(System.in);

System.out.println("Enter all the names:");

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

{

names[i] = s1.nextLine();

}

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

{

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

{

if (names[i].compareTo(names[j])>0)

{

temp = names[i];

names[i] = names[j];

names[j] = temp;

}

}

}

System.out.print("Names in Sorted Order:");

for (int i = 0; i < n - 1; i++)

{

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

}

System.out.print(names[n - 1]);

}

}

1. **WAP using multi-level inheritance.**

class Shape {

   public void display() {

      System.out.println("Inside display");

   }

}

class Rectangle extends Shape {

   public void area() {

      System.out.println("Inside area");

   }

}

class Cube extends Rectangle {

   public void volume() {

      System.out.println("Inside volume");

   }

}

public class Tester {

   public static void main(String[] arguments) {

      Cube cube = new Cube();

      cube.display();

      cube.area();

      cube.volume();

   }

}

1. **WAP to implement hierarchical inheritance.**

class Employee{

float salary = 40000;

}

class PermanentEmp extends Employee{

double hike = 0.5;

}

class TemporaryEmp extends Employee{

double hike = 0.35;

}

class test

{

public static void main(String args[])

{

PermanentEmp p = new PermanentEmp();

TemporaryEmp t = new TemporaryEmp();

System.out.println("Permanent Employee salary is :" +p.salary);

System.out.println("Hike fo r Permanent Employee is:" +p.hike);

System.out.println("Temporary Employee salary is :" +t.salary);

System.out.println("Hike for Temporary Employee is :" +t.hike);

}

}

1. **WAP to implement function overloading to swap 2 numbers.**

public class swap {

public static void main(String[] args) {

int x = 10, y = 20;

char a = 'k', b = 'l';

System.out.println("Before Swapping");

System.out.println("Value of x and y is :" + x + y);

System.out.println("Value of a and b is :" + a + b);

swap(x, y);

swap(a, b);

}

private static void swap(int x, int y) {

int temp = x;

x = y;

y = temp;

System.out.println("After Swapping");

System.out.println("Value of x is :" + x);

System.out.println("Value of y is :" + y);

}

private static void swap(char a, char b) {

char temp = a;

a = b;

b = temp;

System.out.println("After Swapping");

System.out.println("Value of a is :" + a);

System.out.println("Value of b is :" + b);

}

}

1. **WAP to implement abstract class using school student information system.**

import java.util.Scanner;

abstract class Base

{

abstract void fun();

}

class Derived extends Base {

void fun() {

Scanner sc= new Scanner(System.in);

String name= sc.nextLine();

int rollno= sc.nextInt();

String course = sc.nextLine();

float fee= sc.nextFloat();

System.out.println("Student Details: ");

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

System.out.println("Rollno: "+rollno);

System.out.println("Course: "+course);

System.out.println("Fee: "+fee);

}

}

class abs {

public static void main(String args[])

{

Base b = new Derived();

b.fun();

}

}

1. **WAP to implement interfaces using arithmetic calculations i.e., using - addition, subtraction, multiplication, division, remainder - methods.**

interface calculator {

void getresult(int a, int b);

}

class Res implements calculator {

public void getresult(int a, int b)

{

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

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

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

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

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

}

}

class ca {

public static void main(String[] args) {

Res r = new Res();

r.getresult(10, 5);

}

}

1. **WAP to implement library management system using packages.**

**sinfo.java**

package student;

import java.util.\*;

public class sinfo{

public void input(){

Scanner sc=new Scanner(System.in);

System.out.println("Enter Student details");

System.out.println("Enter Student name");

String name=sc.nextLine();

System.out.println("Enter Student course");

String course=sc.nextLine();

System.out.println("Enter rollno");

int rollno=sc.nextInt();

System.out.println("Student Name="+name);

System.out.println("Student rollno="+rollno);

System.out.println("Student course="+course);

}

}

**binfo.java**

package book;

import java.util.\*;

public class binfo{

public void output(){

Scanner sc=new Scanner(System.in);

System.out.println("Enter book details");

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

String bname=sc.nextLine();

System.out.println("Enter author name");

String author=sc.nextLine();

System.out.println("Enter book category");

String type=sc.nextLine();

System.out.println("enter issue date");

String issuedate=sc.nextLine();

System.out.println("Enter return date");

String returndate=sc.nextLine();

System.out.println("Book Name "+bname);

System.out.println("Category "+type);

System.out.println("Author Name=" +author);

System.out.println("Book issue date: "+issuedate);

System.out.println("Book return date: "+returndate);

}

}

**libinfo.java**

package library;

import book.\*;

import student.\*;

class libinfo{

public static void main(String args[])

{

sinfo s=new sinfo();

s.input();

binfo b=new binfo();

b.output();

}

}

1. **WAP to implement student information system using packages.**

**sinfo.java**

package student;

import java.util.\*;

public class sinfo{

public void input(){

Scanner sc=new Scanner(System.in);

System.out.println("Enter Student details");

System.out.println("Enter Student name");

String name=sc.nextLine();

System.out.println("Enter Student course");

String course=sc.nextLine();

System.out.println("Enter rollno");

int rollno=sc.nextInt();

System.out.println("Student Name="+name);

System.out.println("Student rollno="+rollno);

System.out.println("Student course="+course);

}

}

**minfo.java**

package mark;

import java.util.\*;

public class minfo{

public void input()

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter marks of student");

int marks[]=new int[6];

System.out.println("Enter marks of java");

marks[0]=sc.nextInt();

System.out.println("Enter marks of wt");

marks[1]=sc.nextInt();

System.out.println("Enter marks of cd");

marks[2]=sc.nextInt();

System.out.println("Enter marks of cns");

marks[3]=sc.nextInt();

System.out.println("Enter marks of se");

marks[4]=sc.nextInt();

System.out.println("Enter marks of pyscho");

marks[5]=sc.nextInt();

int total=0;

for(int i=0;i<6;i++)

{

total+=marks[i];

}

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

}

}

**stdmark.java**

package Studentinfo;

import student.\*;

import mark.\*;

class stdmark{

public static void main(String args[]){

sinfo s=new sinfo();

s.input();

minfo m=new minfo();

m.input();

}

}

1. **WAP using try, throw and catch.**

class trythrowcat

{

public static void main(String[] args)

{

Scannner sc= new Scanner(System.in);

int a=sc.nextInt();

int b=0,rs;

try

{

if(b==0)

throw(new ArithmeticException("Can't divide by zero."));

else

{

rs = a / b;

System.out.print("\n\tThe result is : " + rs);

}

}

catch(ArithmeticException Ex)

{

System.out.print("\n\tError : " + Ex.getMessage());

}

System.out.print("\n\tEnd of program.");

}

}

1. **WAP using throws and finally (optional - try, throw, catch).**

class ThrowsFin

{

void Division() throws ArithmeticException

{

int a=45,b=0,rs;

rs = a / b;

System.out.print("\n\tThe result is : " + rs);

}

public static void main(String[] args)

{

ThrowsFin T = new ThrowsFin();

try

{

T.Division();

}

catch(ArithmeticException Ex)

{

System.out.print("\n\tError : " + Ex.getMessage());

}

finally

{

System.out.print("\n\tEnd of program.");

}

}

}

1. **WAP using divide by zero error programs.**

class GFG {

    public static void main(String[] args)

    {

        int a = 5;

        int b = 0;

        try {

            System.out.println(a / b);

        }

        catch (ArithmeticException e) {

            System.out.println(

                "Divided by zero operation cannot possible");

        }

    }

}

1. **WAP using array out of bounds exception.**

import java.util.Arrays;

import java.util.Scanner;

public class AOB {

   public static void main(String args[]) {

      int[] myArray = {897, 56, 78, 90, 12, 123, 75};

      System.out.println("Elements in the array are:: ");

      System.out.println(Arrays.toString(myArray));

      Scanner sc = new Scanner(System.in);

      System.out.println("Enter the index of the required element ::");

      try {

         int element = sc.nextInt();

         System.out.println("Element in the given index is :: "+myArray[element]);

      } catch(ArrayIndexOutOfBoundsException e) {

         System.out.println("The index you have entered is invalid");

         System.out.println("Please enter an index number between 0 and 6");

      }

   }

}

1. **WAP using number format exception.**

class nfee {

public static void main(String[] args) {

String s1 = "12";

int i1 = Integer.parseInt(s1);

String s2 = "L";

long l1 = Long.parseLong(s1);

System.out.println(i1 \* l1); // 100

try {

int i3 = Integer.parseInt(s2);

}

catch (NumberFormatException e) {

System.err.println("Unable to format. " + e);

}

}

}

1. **WAP using string index out of bounds exception.**

class SOBE {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the input text : ");

try{

char c = sc.nextLine().charAt(123);

System.out.println("The character at index 123 is : "+c);

}

catch(Exception e)

{

System.out.println("Caught exception : "+e.toString());

}

}

}

1. **Write a program to extend thread class.**

class Class1 extends Thread

{

Class1()

{

super("extending thread");

System.out.println(" thread created" + this);

start();

}

public void run()

{

try

{

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

{

System.out.println("Printing the count " + i);

Thread.sleep(1000);

}

}

catch(InterruptedException e)

{

System.out.println(" thread interrupted");

}

System.out.println("thread run is over" );

}

}

class Extends1

{

public static void main(String args[])

{

Class1 obj = new Class1();

try

{

while(obj.isAlive())

{

System.out.println("Main thread will be alive till the child thread is live");

Thread.sleep(1500);

}

}

catch(InterruptedException e)

{

System.out.println("Main thread interrupted");

}

System.out.println("Main thread's run is over" );

}

}

1. **Program to use join method.**

public class Jo extends Thread{

public void run(){

for(int i=1;i<=5;i++){

try{

Thread.sleep(500);

}

catch(Exception e)

{

System.out.println(e);

}

System.out.println(i);

}

}

public static void main(String args[]){

Jo t1=new Jo();

Jo t2=new Jo();

Jo t3=new Jo();

t1.start();

try{

t1.join();

}

catch(Exception e)

{

System.out.println(e);

}

t2.start();

t3.start();

}

}

1. **Program using non-synchronized threads.**

class Sync1{

void method1(int n){//method not synchronized

for(int i=1;i<=5;i++){

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

try{

Thread.sleep(1000);

}

catch(Exception e)

{

System.out.println(e); }

}

}

}

class Thread1 extends Thread{

Sync1 t;

Thread1(Sync1 t){

this.t=t;

}

public void run(){

t.method1(5);

}

}

class Thread2 extends Thread{

Sync1 t;

Thread2(Sync1 t){

this.t=t;

}

public void run(){

t.method1(100);

}

}

class synchro1{

public static void main(String args[]){

Sync1 obj = new Sync1();//only one object

Thread1 t1=new Thread1(obj);

Thread2 t2=new Thread2(obj);

t1.start();

t2.start();

}

}

1. **Program to implement interthread communication.**

class Customer{

int amount=10000;

synchronized void withdraw(int amount){

System.out.println("going to withdraw...");

if(this.amount<amount){

System.out.println("Less balance; waiting for deposit...");

try{

wait();

}

catch(Exception e){}

}

this.amount-=amount;

System.out.println("withdraw completed...");

}

synchronized void deposit(int amount){

System.out.println("going to deposit...");

this.amount+=amount;

System.out.println("deposit completed... ");

notify();

}

}

class Test{

public static void main(String[] args){

final Customer c=new Customer();

new Thread()

{

public void run(){

c.withdraw(15000);

}

}.start();

new Thread(){

public void run()

{

c.deposit(10000);}

}.start();

}}

1. **Program using deadlock.**

public class deadl {

public static void main(String[] args) {

final String resource1 = "abc";

final String resource2 = "xyz";

Thread t1 = new Thread() {

public void run()

{

synchronized (resource1)

{

System.out.println("Thread 1: locked resource 1");

try

{

Thread.sleep(100);

}

catch (Exception e) { }

synchronized (resource2) {

System.out.println("Thread 1: locked resource 2");

}

}

}

};

Thread t2 = new Thread() {

public void run() {

synchronized (resource2) {

System.out.println("Thread 2: locked resource 2");

try { Thread.sleep(100);} catch (Exception e) {}

synchronized (resource1) {

System.out.println("Thread 2: locked resource 1");

}

}

}

};

t1.start();

t2.start();

}

}

1. **Program using synchronized threads.**

class Table{

synchronized static void method1(int n){

for(int i=1;i<=5;i++){

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

try{

Thread.sleep(400);

}

catch(Exception e){}

}

}

}

class Thread1 extends Thread{

public void run(){

Table.method1(1);

}

}

class Thread2 extends Thread{

public void run(){

Table.method1(10);

}

}

class Thread3 extends Thread{

public void run(){

Table.method1(100);

}

}

class Thread4 extends Thread{

public void run(){

Table.method1(1000);

}

}

public class synchro3{

public static void main(String t[]){

Thread1 t1=new Thread1();

Thread2 t2=new Thread2();

Thread3 t3=new Thread3();

Thread4 t4=new Thread4();

t1.start();

t2.start();

t3.start();

t4.start();

}

}

1. **Program using thread priority.**

public class prior extends Thread

{

public static void main(String[] args)

{

prior thread1 = new prior ();

prior thread2 = new prior ();

prior thread3 = new prior ();

System.out.println("The priority of thread1 = " + thread1.getPriority());

System.out.println("The priority of thread2 = " + thread2.getPriority());

System.out.println("The priority of thread3 = " + thread3.getPriority());

thread1.setPriority(6);

thread2.setPriority(7);

thread3.setPriority(8);

System.out.println("The priority of thread1 = " + thread1.getPriority());

System.out.println("The priority of thread2 = " + thread2.getPriority());

System.out.println("The priority of thread3 = " + thread3.getPriority());

System.out.println("The priority of Main thread = " +

Thread.currentThread().getPriority());

Thread.currentThread().setPriority(10);

System.out.println("The priority of Main thread = " +

Thread.currentThread().getPriority());

}

}

1. **Write a Java program for handling mouse events.**

import java.awt.\*;

import java.awt.event.\*;

public class mouse extends Frame implements MouseListener{

Label l;

mouse(){

addMouseListener(this);

l=new Label();

l.setBounds(20,50,100,20);

add(l);

setSize(300,300);

setLayout(null);

setVisible(true);

}

public void mouseClicked(MouseEvent e) {

l.setText("Mouse Clicked");

}

public void mouseEntered(MouseEvent e) {

l.setText("Mouse Entered");

}

public void mouseExited(MouseEvent e) {

l.setText("Mouse Exited");

}

public void mousePressed(MouseEvent e) {

l.setText("Mouse Pressed");

}

public void mouseReleased(MouseEvent e) {

l.setText("Mouse Released");

}

public static void main(String[] args) {

new mouse();

}

}

1. **Write a program using key events.**

import java.awt.\*;

import java.awt.event.\*;

public class key extends Frame implements KeyListener{

Label l;

TextArea area;

key(){

l=new Label();

l.setBounds(20,50,100,20);

area=new TextArea();

area.setBounds(20,80,300, 300);

area.addKeyListener(this);

add(l);

add(area);

setSize(400,400);

setLayout(null);

setVisible(true);

}

public void keyPressed(KeyEvent e) {

l.setText("Key Pressed");

}

public void keyReleased(KeyEvent e) {

l.setText("Key Released");

}

public void keyTyped(KeyEvent e) {

l.setText("Key Typed");

}

public static void main(String[] args) {

new key();

}

}

1. **Program using runnable interface.**

class Class2 implements Runnable

{

Thread mythread ;

Class2()

{

mythread = new Thread(this, "my runnable thread");

System.out.println("my thread created" + mythread);

mythread.start();

}

public void run()

{

try

{

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

{

System.out.println("Printing the count " + i);

Thread.sleep(1000);

}

}

catch(InterruptedException e)

{

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

}

System.out.println("Thread run is over" );

}

}

class Runnable1

{

public static void main(String args[])

{

Class2 obj = new Class2();

try

{

while(obj.mythread.isAlive())

{

System.out.println("Main thread will be alive till the child thread is live");

Thread.sleep(1500);

}

}

catch(InterruptedException e)

{

System.out.println("Main thread interrupted");

}

System.out.println("Main thread run is over" );

}

}

1. **Write a java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green. When a radio button is selected, the light is turned on, and only one light can be on at a time No light is on when the program starts.**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

class TrafficLight extends JFrame implements ActionListener

{

String msg=" " ;

private JLabel label;

private JTextField display;

private JRadioButton r1,r2,r3;

private ButtonGroup bg;

private Container c;

public TrafficLight()

{

setLayout(new FlowLayout());

c=getContentPane();

label=new JLabel("Traffic Light");

display =new JTextField(10);

r1=new JRadioButton("RED");

r2=new JRadioButton("GREEN");

r3=new JRadioButton("YELLOW");

bg=new ButtonGroup();

c.add(label);

c.add(r1);

c.add(r2);

c.add(r3);

c.add(display);

bg.add(r1);

bg.add(r2);

bg.add(r3);

r1.addActionListener(this);

r2.addActionListener(this);

r3.addActionListener(this);

setSize(400,400);

setVisible(true);

c.setBackground(Color.white);

}

public void actionPerformed(ActionEvent ie)

{

msg=ie.getActionCommand();

if (msg.equals("RED"))

{

c.setBackground(Color.RED);

display.setText(msg+ " :TURN ON");

}

else if (msg.equals("GREEN"))

{

c.setBackground(Color.GREEN);

display.setText(msg+ " :TURN ON");

}

else if (msg.equals("YELLOW"))

{

c.setBackground(Color.YELLOW);

display.setText(msg+ ":TURN ON");

}

}

public static void main(String args[])

{

TrafficLight light=new TrafficLight();

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

}

}

1. **Write all programs using Layout Managers.**
2. **BorderLayout B) GridLayout C) FlowLayout D) CardLayout D) GridBagLayout**

**BorderLayout:**

import java.awt.\*;

import javax.swing.\*;

class Border {

JFrame f;

Border(){

f=new JFrame();

JButton b1=new JButton("NORTH");;

JButton b2=new JButton("SOUTH");;

JButton b3=new JButton("EAST");;

JButton b4=new JButton("WEST");;

JButton b5=new JButton("CENTER");;

f.add(b1,BorderLayout.NORTH);

f.add(b2,BorderLayout.SOUTH);

f.add(b3,BorderLayout.EAST);

f.add(b4,BorderLayout.WEST);

f.add(b5,BorderLayout.CENTER);

f.setSize(300,300);

f.setVisible(true);

}

public static void main(String[] args) {

new Border();

}

}

**GridLayout:**

import java.awt.\*;

import javax.swing.\*;

class gl{

JFrame f;

gl(){

f=new JFrame();

JButton b1=new JButton("1");

JButton b2=new JButton("2");

JButton b3=new JButton("3");

JButton b4=new JButton("4");

JButton b5=new JButton("5");

JButton b6=new JButton("6");

JButton b7=new JButton("7");

JButton b8=new JButton("8");

JButton b9=new JButton("9");

f.add(b1);f.add(b2);f.add(b3);f.add(b4);f.add(b5);

f.add(b6);f.add(b7);f.add(b8);f.add(b9);

f.setLayout(new GridLayout(3,3));

f.setSize(300,300);

f.setVisible(true);

}

public static void main(String[] args) {

new gl();

}

}

**FlowLayout:**

import java.awt.\*;

import javax.swing.\*;

classfl{

JFrame f;

fl(){

f=new JFrame();

JButton b1=new JButton("1");

JButton b2=new JButton("2");

JButton b3=new JButton("3");

JButton b4=new JButton("4");

JButton b5=new JButton("5");

f.add(b1);f.add(b2);f.add(b3);f.add(b4);f.add(b5);

f.setLayout(new FlowLayout(FlowLayout.RIGHT));

f.setSize(300,300);

f.setVisible(true);

}

public static void main(String[] args) {

new fl();

}

}

**CardLayout:**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

class CardLayoutExample extends JFrame implements ActionListener{

CardLayout card;

JButton b1,b2,b3;

Container c;

CardLayoutExample(){

c=getContentPane();

card=new CardLayout(40,30);

c.setLayout(card);

b1=new JButton("Apple");

b2=new JButton("Boy");

b3=new JButton("Cat");

b1.addActionListener(this);

b2.addActionListener(this);

b3.addActionListener(this);

c.add("a",b1);c.add("b",b2);c.add("c",b3);

}

public void actionPerformed(ActionEvent e) {

card.next(c);

}

public static void main(String[] args) {

CardLayoutExample cl=new CardLayoutExample();

cl.setSize(400,400);

cl.setVisible(true);

cl.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

}

}

**GridBagLayout:**

import java.awt.\*;

import javax.swing.\*;

class GBL extends JFrame{

public static void main(String[] args) {

GBL a = new GBL();

}

Public GBL() {

GridBagLayout grid = new GridBagLayout();

GridBagConstraints gbc = new GridBagConstraints();

setLayout(grid);

setTitle("GridBag Layout Example");

GridBagLayout layout = new GridBagLayout();

this.setLayout(layout);

gbc.fill = GridBagConstraints.HORIZONTAL;

gbc.gridx = 0;

gbc.gridy = 0;

this.add(new Button("Button One"), gbc);

gbc.gridx = 1;

gbc.gridy = 0;

this.add(new Button("Button two"), gbc);

gbc.fill = GridBagConstraints.HORIZONTAL;

gbc.ipady = 20;

gbc.gridx = 0;

gbc.gridy = 1;

this.add(new Button("Button Three"), gbc);

gbc.gridx = 1;

gbc.gridy = 1;

this.add(new Button("Button Four"), gbc);

gbc.gridx = 0;

gbc.gridy = 2;

gbc.fill = GridBagConstraints.HORIZONTAL;

gbc.gridwidth = 2;

this.add(new Button("Button Five"), gbc);

setSize(300, 300);

setPreferredSize(getSize());

setVisible(true);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

}

}

1. **Write a program using applet life cycle.**

import java.applet.Applet;

import java.awt.Graphics;

import java.awt.\*;

public class AppletLifeCycle extends Applet

{

public void init()

{

setBackground(Color.CYAN);

System.out.println("init() called");

}

public void start(){

System.out.println("Start() called");

}

public void paint(Graphics g){

System.out.println("Paint() called");

}

public void stop() {

System.out.println("Stop() Called");

}

public void destroy() {

System.out.println("Destroy)() Called");

}

}

**AppletLifeCycle.html**

<html>

<applet code="AppletLifeCycle" width="350" height="150">

</applet>

</html>

1. **Creating Frames using Swings in Java (Creating the object of Frame class (association))**

import javax.swing.\*;

import javax.swing.\*;

import java.awt.\*;

class fs

{

JFrame jf;

public fs()

{

jf = new JFrame("MyWindow");

JButton btn = new JButton(" Hello Swings”);

jf.add(btn);

jf.setLayout(new FlowLayout());

jf.setSize(400, 400);

jf.setVisible(true);

}

public static void main(String[] args)

{

new fs();

}

}

1. **Creating Frames using Swings in Java (By extending Frame class (inheritance))**

import javax.swing.\*;

import java.awt.\*;

class Second extends JFrame

{

public Second()

{

setTitle("MyWindow");

JLabel lb = new JLabel("Welcome to My Second Window");

add(lb);

setLayout(new FlowLayout());

setSize(400, 400);

setVisible(true);

}

public static void main(String[] args)

{

new Second();

}

}