**JAVA PROGRAMS**

**0**

class hello

{

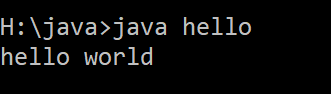
public static void main(String args[])

{

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

}

}



**1**

//examples of integers and float values:

class datatypes

{

public static void main(String args[])

{

int a=10;

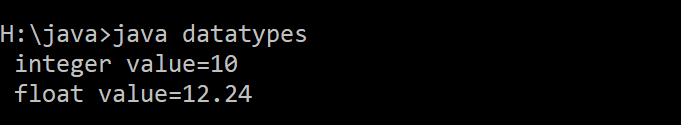
float b=12.24f;

System.out.println(" integer value="+a);

System.out.println(" float value="+b);

}

}



**2**

// add two numbers:

import java.util.Scanner;

class addition

{

public static void main(String args[])

{

Scanner key=new Scanner(System.in);

int num1,num2,sum;

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

num1=key.nextInt();

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

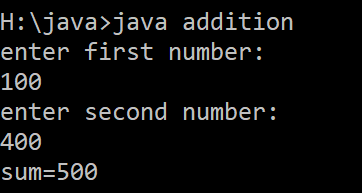
num2=key.nextInt();

sum=num1+num2;

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

}

}



**3**

//input string value:

import java.util.Scanner;

class string

{

public static void main(String args[])

{

Scanner key=new Scanner(System.in);

String name;

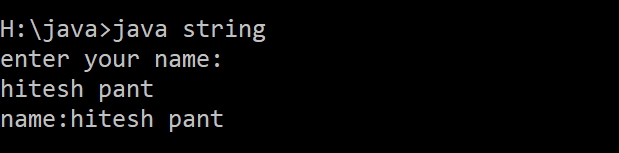
System.out.println("enter your name:");

name=key.nextLine();

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

}

}



**4**

//use of conditional statements(if-else):

import java.util.Scanner;

class if\_else

{

public static void main(String args[])

{

Scanner key=new Scanner(System.in);

int age;

System.out.println("enter your age:");

age=key.nextInt();

if(age<18)

{

System.out.println("not eligible");

}

else

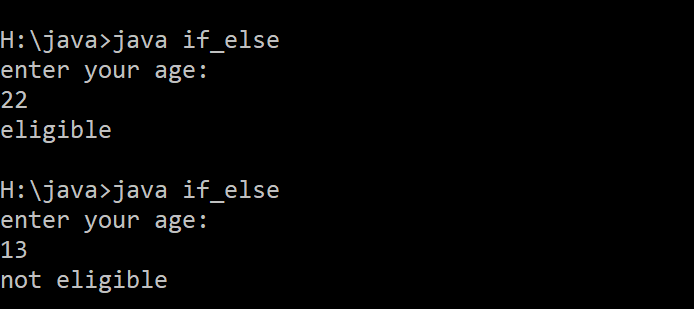
{

System.out.println("eligible");

}

}

}



**5**

//illustration of while loop:

class whileloop

{

public static void main(String args[])

{

int i=0;

while(i<=10)

{

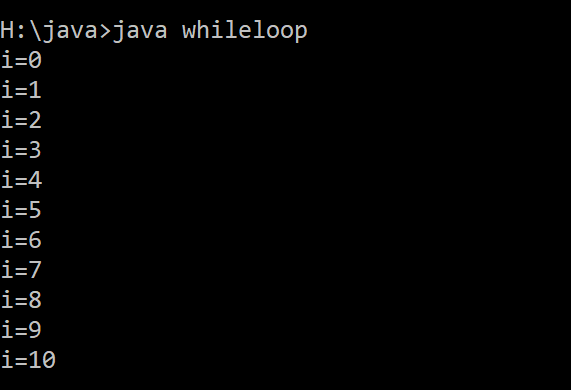
System.out.println("i="+i);

i++;

}

}

}



**6**

import java.util.Scanner;

class marksheet

{

public static void main(String args[])

{

Scanner key=new Scanner(System.in);

String name,division;

int roll,h,e,sc,m,total,perc;

System.out.println("\t\t\t STUDENT DETAILS");

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

System.out.println("enter student name:");

name=key.nextLine();

System.out.println("enter sudent roll.no:");

roll=key.nextInt();

System.out.println("enter hindi marks:");

h=key.nextInt();

System.out.println("enter english marks:");

e=key.nextInt();

System.out.println("enter science marks:");

sc=key.nextInt();

System.out.println("enter maths marks:");

m=key.nextInt();

System.out.println("\t\t\t UNIVERSITY OF MADHYA PRADESH");

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

System.out.println("NAME:"+name+"\t\t\t ROLL NUMBER:"+roll);

System.out.println(" ");

System.out.println("SUBJECT NAME \t\t MAX \t\t MIN \t\t OBT");

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

System.out.println("HINDI \t\t\t100 \t\t33 \t\t"+h);

System.out.println("ENGLISH \t\t100 \t\t33 \t\t"+e);

System.out.println("SCIENCE \t\t100 \t\t33 \t\t"+sc);

System.out.println("MATHS \t\t\t100 \t\t33 \t\t"+m);

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

total=(h+e+sc+m);

perc=total/4;

System.out.println("TOTAL \t\t\t\t\t\t\t"+total);

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

System.out.println("PERCENTAGE \t\t\t\t\t\t"+perc);

if(perc>=50)

{

System.out.println("RESULT \t\t\t\t\t\t pass");

}

else

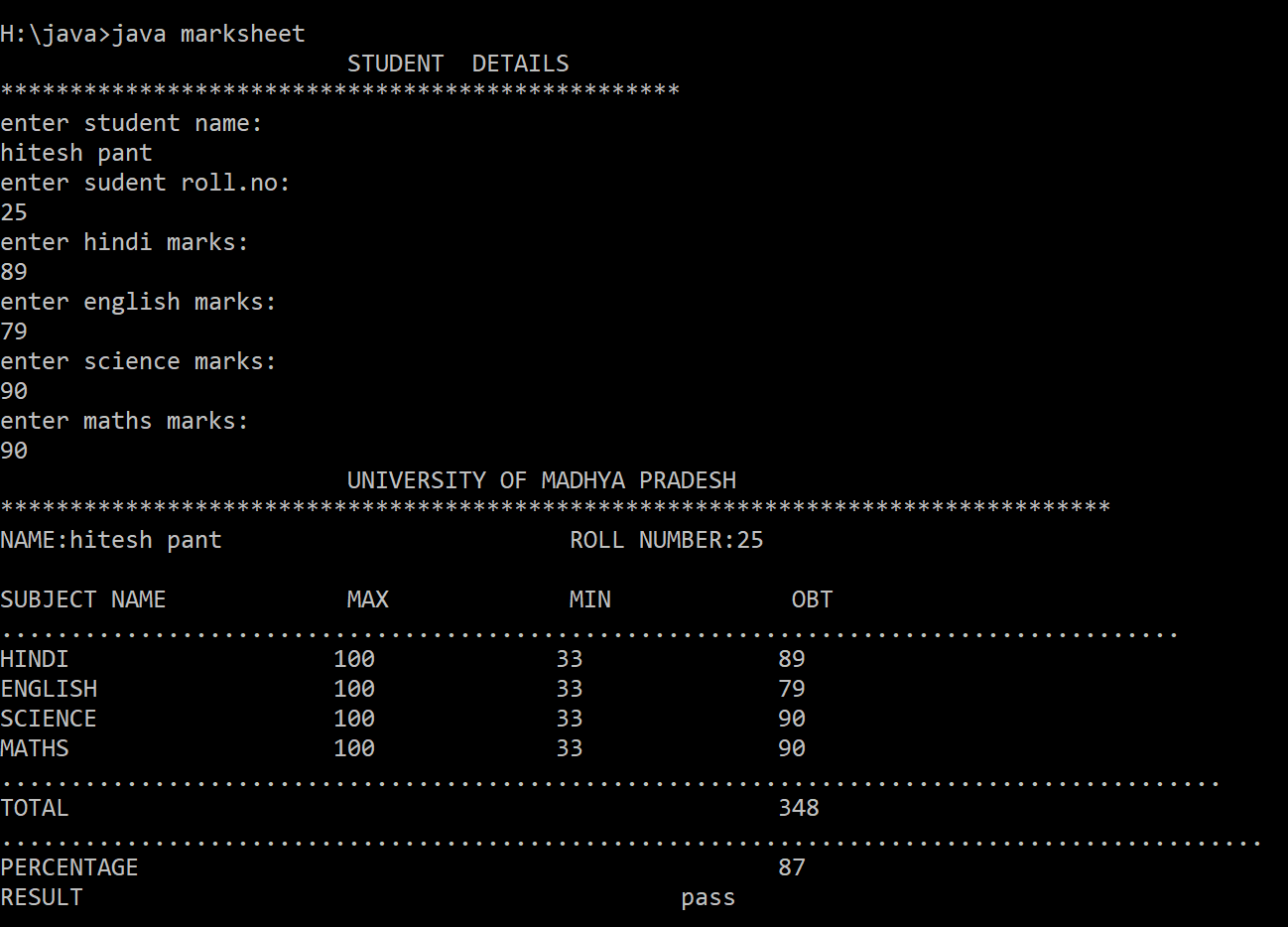
{

System.out.println("RESULT \t\t\t\t\t\t fail");

}

}

}



**7**

// program to check whether a charatcer is vowel or consonant:

import java.util.Scanner;

class switchcase

{

public static void main(String args[])

{

Scanner key=new Scanner(System.in);

char ch;

System.out.println("enter any character:");

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

switch(ch)

{

case 'a':

System.out.println(ch+" is a vowel");

break;

case 'e':

System.out.println(ch+" is a vowel");

break;

case 'i':

System.out.println(ch+" is a vowel");

break;

case 'o':

System.out.println(ch+" is a vowel");

break;

case 'u':

System.out.println(ch+" is a vowel");

break;

default:

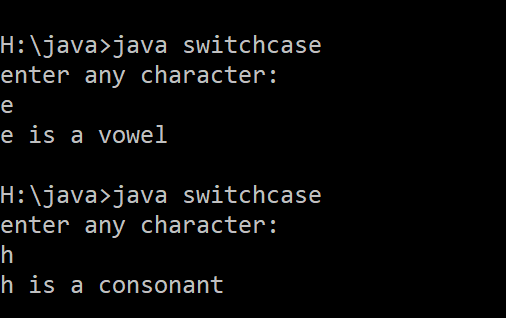
System.out.println(ch+" is a consonant");

break;

}

}

}



**8**

//program to add two numbers using function with no argumentns and no returntype:

import java.util.Scanner;

class function1

{

public static void add()

{

Scanner key=new Scanner(System.in);

int num1,num2,sum;

System.out.println("enter two numbers:");

num1=key.nextInt();

num2=key.nextInt();

sum=num1+num2;

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

}

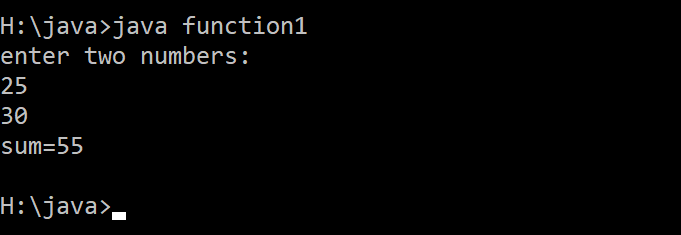
public static void main(String args[])

{

add();

}

}



**9**

//pprigram to add 2 numbers using function accepting arguments with returning value:

import java.util.Scanner;

class function2

{

public static int add(int num1,int num2)

{

int sum;

sum=num1+num2;

return(sum);

}

public static void main(String args[])

{

Scanner key=new Scanner(System.in);

int num1,num2;

System.out.println("enter 2 numbers:");

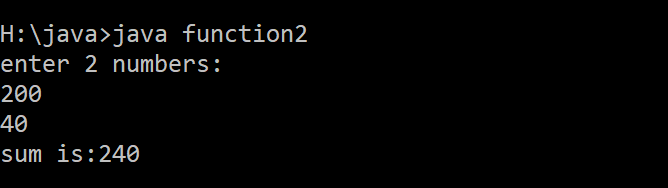
num1=key.nextInt();

num2=key.nextInt();

System.out.println("sum is:"+add(num1,num2));

}

}



**10**

//perform addition,subtraction,multiplication,division of two numbers:

import java.util.Scanner;

class operations

{

public static void main(String args[])

{

Scanner key=new Scanner(System.in);

int num1,num2,sum,sub,mul,div;

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

num1=key.nextInt();

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

num2=key.nextInt();

sum=num1+num2;

sub=num1-num2;

mul=num1\*num2;

div=num1/num2;

System.out.println("addition:"+sum);

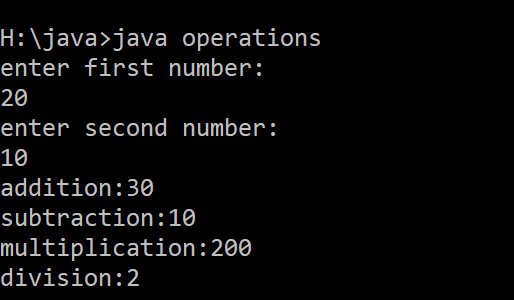
System.out.println("subtraction:"+sub);

System.out.println("multiplication:"+mul);

System.out.println("division:"+div);

}

}



**11**

//CONSTRUCTOR:

class student

{

student()

{

System.out.println("this is a constructor");

}

}

class constructor1

{

public static void main(String args[])

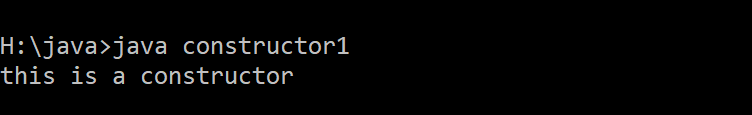
{

student obj=new student();

//it is not necessary to invoke construcor using object.

}

}



**12**

// add two numbers using constructor:

import java.util.Scanner;

class add

{

add()

{

Scanner key=new Scanner(System.in);

int num1,num2,sum;

System.out.println("enter two numbers");

num1=key.nextInt();

num2=key.nextInt();

sum=num1+num2;

System.out.println("sum is:"+sum);

}

}

class constructor2

{

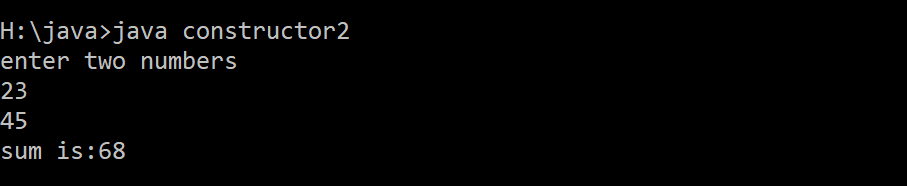
public static void main(String args[])

{

add obj=new add();

}

}



**13**

// add two matrix:

import java.util.Scanner;

class addmatrix

{

public static void main(String args[])

{

Scanner key=new Scanner(System.in);

int arr1[][];

int arr2[][];

int arr3[][];

arr1=new int[2][2];

arr2=new int[2][2];

arr3=new int[2][2];

System.out.println("enter numbers in matrix 1:");

arr1[0][0]=key.nextInt();

arr1[0][1]=key.nextInt();

arr1[1][0]=key.nextInt();

arr1[1][1]=key.nextInt();

System.out.println("enter numbers in matrix 2:");

for(int i=0;i<2;i++)

{

for(int j=0;j<2;j++)

{

arr2[i][j]=key.nextInt();

}

}

arr3[0][0]=arr1[0][0]+arr2[0][0];

arr3[0][1]=arr1[0][1]+arr2[0][1];

arr3[1][0]=arr1[1][0]+arr2[1][0];

arr3[1][1]=arr1[1][1]+arr2[1][1];

System.out.println("sum of two matrices is:");

System.out.print(arr3[0][0]);

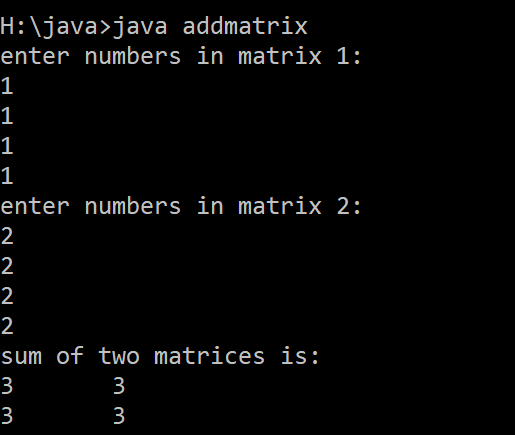
System.out.print("\t"+arr3[0][1]);

System.out.print("\n"+arr3[1][0]);

System.out.print("\t"+arr3[1][1]);

}

}



**14**

// single level inheritance:

class A

{

public static void fun1()

{

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

}

}

class B extends A

{

public static void fun2()

{

System.out.println("class B");

}

}

class single\_inheritance

{

public static void main(String args[])

{

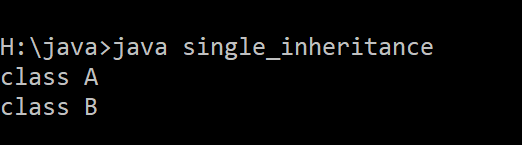
B obj1=new B();

obj1.fun1();

obj1.fun2();

}

}



**15**

//multi-level inheritance:

class A

{

public static void fun1()

{

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

}

}

class B extends A

{

public static void fun2()

{

System.out.println("class B");

}

}

class C extends B

{

public static void fun3()

{

System.out.println("class C");

}

}

class multilevel\_inhertiance

{

public static void main(String args[])

{

C obj=new C();

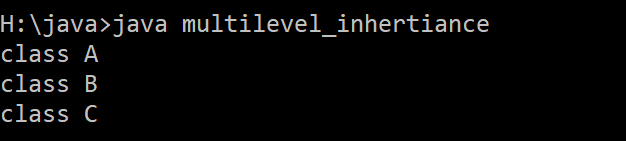
obj.fun1();

obj.fun2();

obj.fun3();

}

}



**16**

//illustrate hierarchical inhertiance:

class A

{

public static void fun1()

{

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

}

}

class B extends A

{

public static void fun2()

{

System.out.println("class B");

}

}

class C extends A

{

public static void fun3()

{

System.out.println("class C");

}

}

class hierarchical\_inhertiance

{

public static void main(String args[])

{

C obj1=new C();

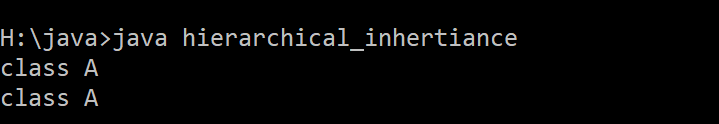
B obj2=new B();

obj1.fun1();

obj2.fun1();

}

}



**17**

// HYBRID INHERITANCE:

class A

{

public static void fun1()

{

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

}

}

class B extends A

{

public static void fun2()

{

System.out.println("class B");

}

}

class C extends B

{

public static void fun3()

{

System.out.println("class C");

}

}

class D extends B

{

public static void fun4()

{

System.out.println("class D");

}

}

class hybrid\_inhertiance

{

public static void main(String args[])

{

B obj1=new B();

C obj2=new C();

D obj3=new D();

obj1.fun1();

obj2.fun2();

obj2.fun2();

obj3.fun2();

}

}

**18**

//interface:

interface interface1

{

void print();

}

class interface11 implements interface1

{

public void print()

{

System.out.println("this is an interface named 'interface1'");

}

public static void main(String args[])

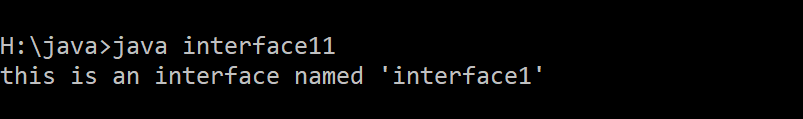
{

interface11 obj=new interface11();

obj.print();

}

}



**19**

//interface:

interface drawable

{

void draw();

}

//implementation by user:

class rectangle implements drawable

{

public void draw()

{

System.out.println("drawing rectangle");

}

}

class circle implements drawable

{

public void draw()

{

System.out.println("drawing circle");

}

}

class interface22 //main class

{

public static void main(String args[])

{

drawable obj1=new circle();

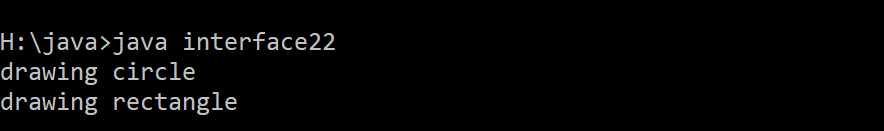
drawable obj2=new rectangle();

obj1.draw();

obj2.draw();

}

}



**20**

//interface:

interface bank

{

float interestrate();

}

class SBI implements bank

{

public float interestrate()

{

return(9.15f);

}

}

class PNB implements bank

{

public float interestrate()

{

return(4.2f);

}

}

class interface33

{

public static void main(String args[])

{

bank obj1=new SBI();

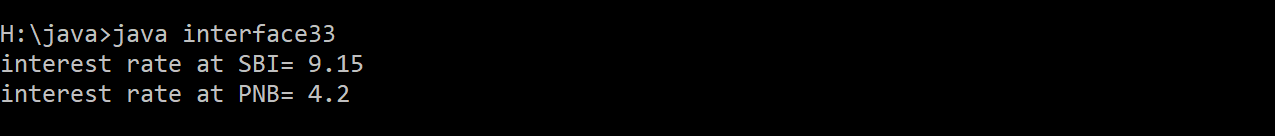
System.out.println("interest rate at SBI= "+obj1.interestrate());

bank obj2=new PNB();

System.out.println("interest rate at PNB= "+obj2.interestrate());

}

}



**21**

//PACKAGE:

package package1;

public class A

{

public void show()

{

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

}

}

//implementaion of package1:

import package1.\*;

class packagetest1

{

public static void main(String args[])

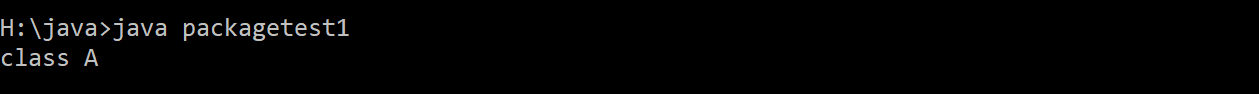
{

A obj=new A();

obj.show();

}

}



**22**

// package:

package package2;

public class p2

{

public void fun1()

{

System.out.println("fun1 in package2");

}

public void fun2()

{

System.out.println("fun2 in package2");

}

}

// implement package2 :

import package2.\*;

class packagetest2

{

public static void main(String args[])

{

p2 obj1=new p2();

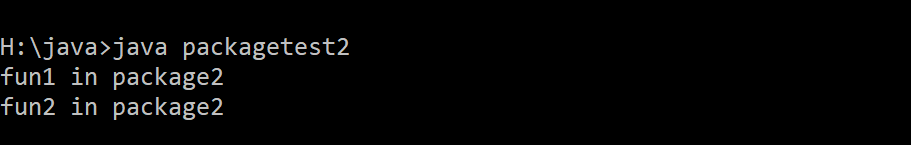
p2 obj2=new p2();

obj1.fun1();

obj2.fun2();

}

}



**23**

// package:

package package3;

public class p3

{

public void fun1()

{

System.out.println("fun1 in package3");

}

}

import package1.\*;

import package2.\*;

import package3.\*;

class packagetest3

{

public static void main(String args[])

{

A obj=new A();

p2 obj1=new p2();

p2 obj2=new p2();

p3 obj4=new p3();

obj.show();

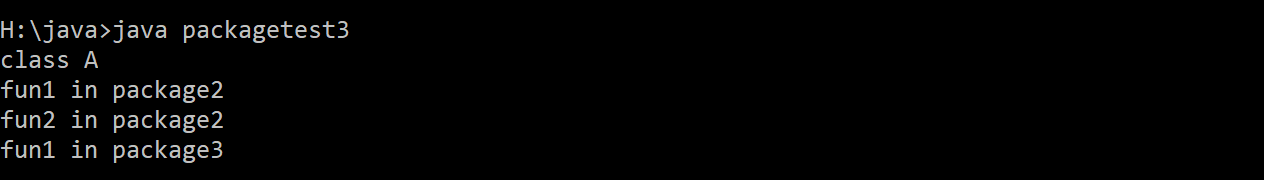
obj1.fun1();

obj2.fun2();

obj4.fun1();

}

}



**24**

import java.applet.Applet;

import java.awt.Graphics;

//<applet code="applet1.class"width="300"height="300"></applet>

public class applet1 extends Applet

{

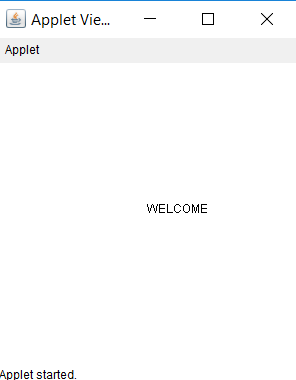
public void paint(Graphics g)

{

g.drawString("WELCOME",150,150);

}

}



**25**

// Buttons in applet:

import java.awt.\*;

import java.applet.\*;;

//<applet code="applet\_button1.class"width="300"height="300"></applet>

public class applet\_button1 extends Applet

{

Button btn=new Button("OK");

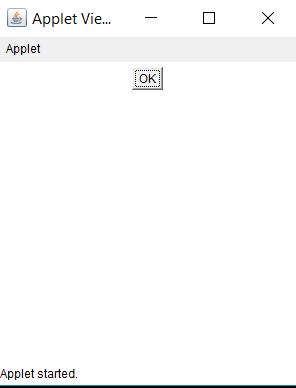
public void init()

{

add(btn);

}

}



**26**

// textfield in applet:

import java.awt.\*;

import java.applet.\*;

//<applet code="applet\_textfield1.class"width="400"height="400"></applet>

public class applet\_textfield1 extends Applet

{

TextField t1=new TextField(30);

TextField t2=new TextField(30);

public void init()

{

add(t1);

add(t2);

}

}



**27**

// buttons and textfields to perform some task:

import java.awt.\*;

import java.applet.\*;

import java.awt.event.\*;

//<applet code="applet\_project1.class"width="500"height="500"></applet>b

public class applet\_project1 extends Applet implements ActionListener

{

TextField t1=new TextField(20);

TextField t2=new TextField(20);

TextField t3=new TextField(20);

Button b1=new Button("click here");

Button b2=new Button("click here");

Button b3=new Button("click here");

public void init()

{

b1.addActionListener(this);

b2.addActionListener(this);

b3.addActionListener(this);

add(t1);

add(b1);

add(t2);

add(b2);

add(t3);

add(b3);

}

public void actionPerformed(ActionEvent ae)

{

if(ae.getSource()==b1)

{

t1.setText("hello");

}

if(ae.getSource()==b2)

{

t2.setText("welcome");

}

if(ae.getSource()==b3)

{

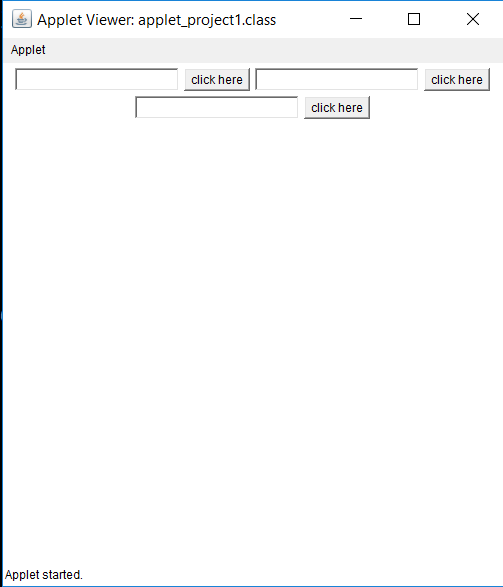
t3.setText("bye");

}

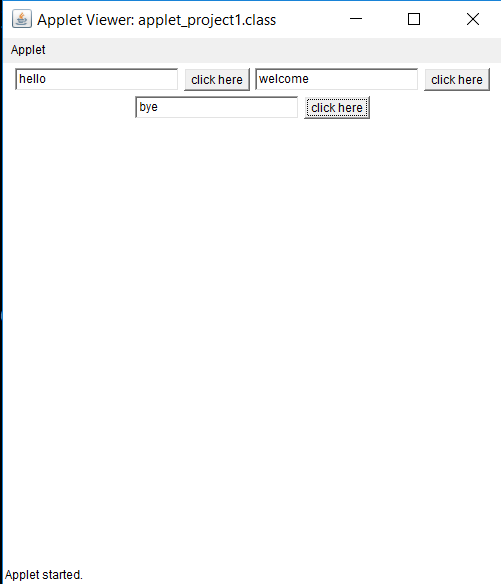
}

}

Output before:



**Output after:**



**28**

// CREATING THREADS:

class A extends Thread

{

public void run()

{

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

{

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

}

}

}

class B extends Thread

{

public void run()

{

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

{

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

}

}

}

class C extends Thread

{

public void run()

{

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

{

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

}

}

}

class threads

{

public static void main(String args[])

{

A obj1=new A();

B obj2=new B();

C obj3=new C();

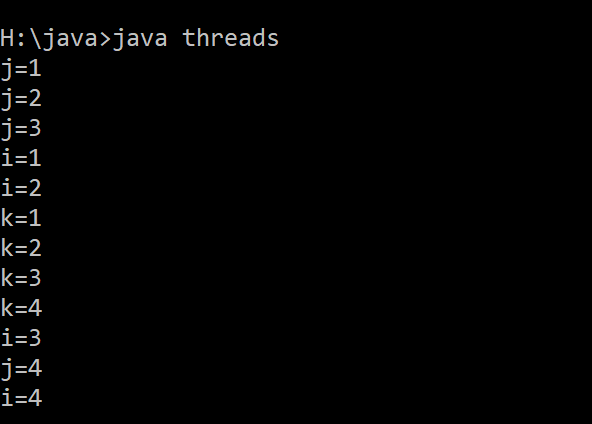
obj1.start();

obj2.start();

obj3.start();

}

}



**29**

//use of isAlive() and join in multithreading:

class A extends Thread

{

public void run()

{

System.out.println("status:"+isAlive());

}

}

class isAlive\_JoinInThread

{

public static void main(String args[])

{

A obj=new A();

obj.start();

try

{

obj.join(); //join() waits for a thread to die,it stop executing current thread.

}

catch(InterruptedException e)

{

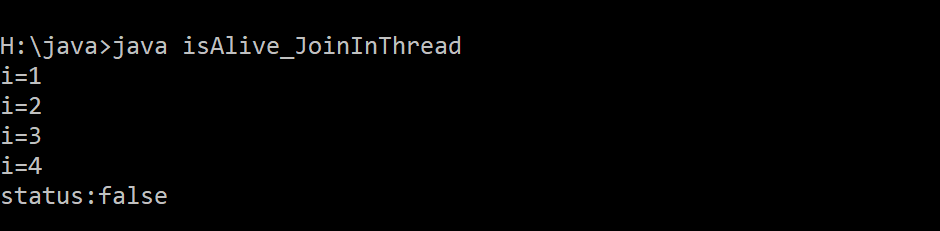
}

System.out.println("status:"+obj.isAlive()); //isAlive() checks the thread is alive or not.

}

}

// A thread runs only for one time then it dies itself.



**30**

//priority in thread:

class A extends Thread

{

public void run()

{

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

{

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

}

}

}

class B extends Thread

{

public void run()

{

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

{

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

}

}

}

class PriorityInThread

{

public static void main(String args[])

{

A obj1=new A();

B obj2=new B();

obj1.setPriority(10);

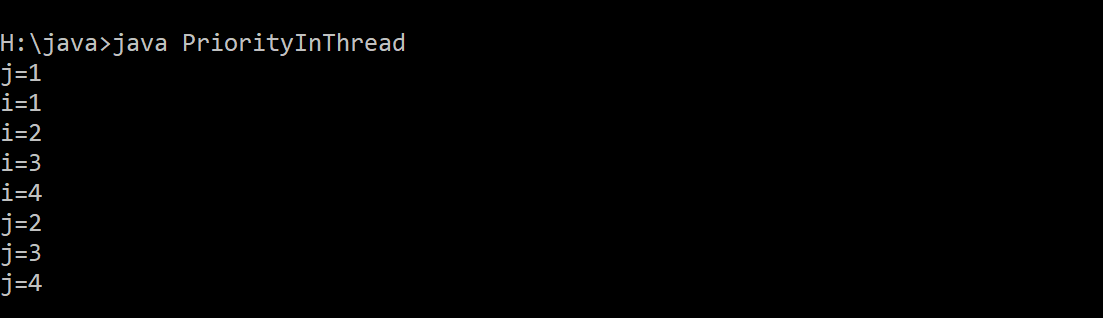
obj2.setPriority(1);

obj1.start();

obj2.start();

}

}



**31**

class A extends Thread

{

public void run()

{

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

{

if(i==3)

{

try

{

sleep(5000);

}

catch(Exception e)

{

}

}

}

}

}

class B extends Thread

{

public void run()

{

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

{

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

}

}

}

class SleepInThread

{

public static void main(String args[])

{

A obj1=new A();

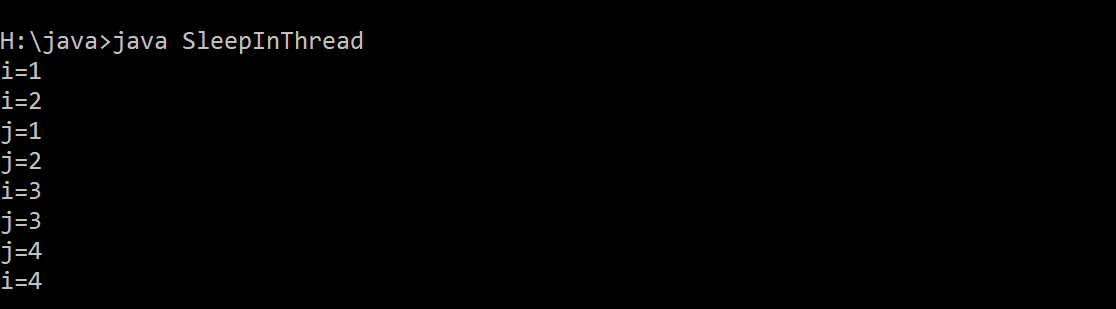
B obj2=new B();

obj1.start();

obj2.start();

}

}



**32**

//stop() in thread:

class A extends Thread

{

public void run()

{

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

{

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

}

}

}

class B extends Thread

{

public void run()

{

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

{

if(j==3)

{

stop(); // put in dead condtition

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

}

}

}

}

class C extends Thread

{

public void run()

{

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

{

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

}

}

}

class StopInThread

{

public static void main(String args[])

{

A obj1=new A();

B obj2=new B();

C obj3=new C();

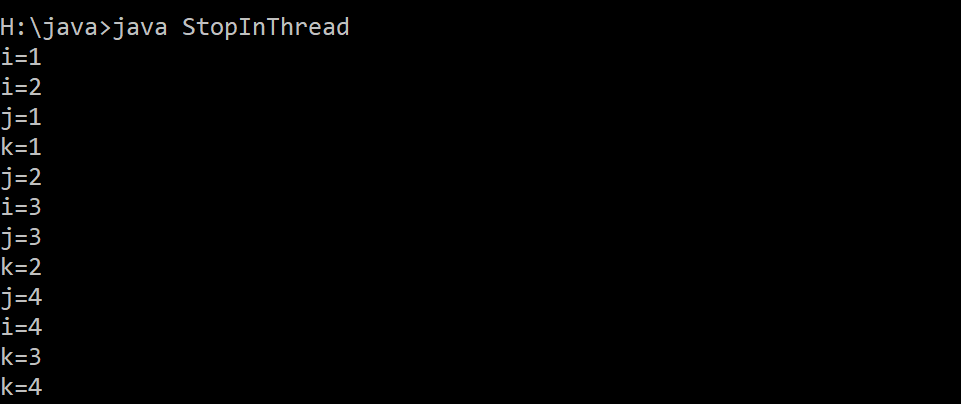
obj1.start();

obj2.start();

obj3.start();

}

}



**32**

//yield in thread:

class A extends Thread

{

public void run()

{

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

{

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

}

}

}

class B extends Thread

{

public void run()

{

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

{

if(k==2)

{

yield();

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

}

}

}

}

class C extends Thread

{

public void run()

{

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

{

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

}

}

}

class YieldInThread

{

public static void main(String args[])

{

A obj1=new A();

B obj2=new B();

C obj3=new C();

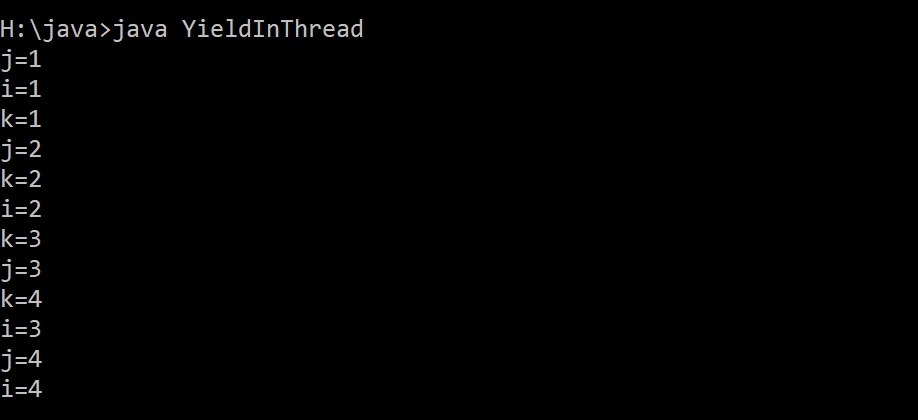
obj1.start();

obj2.start();

obj3.start();

}

}



**33**

//synchronozation in multithreading:

class update

{

Synchronized void updateSum(int i)

{

Thread t=Thread.currentThread();

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

{

System.out.println(t.getname()+":"+(1+n));

}

}

}

class A extends Thread

{

update u=new update();

public void run()

{

u.updatesum(10);

}

}

class SynchronizedThread

{

public static void main(String args[])

{

A a=new A();

Thread t1=new Thread(a);

Thread t2=new Thread(a);

t1.setName("Thread A");

t2.setName("thread B");

t1.start();

t2.start();

}

}