**LAB REPORT**

**ON**

**java PROGRAMMING**

**Phase II**

**By**

**Rupesh mahat**

**Exam Roll No: 8659/18**



**Submitted to:**

**Nimesh Pokhrel**

**Department of Computer Science**

**Kantipur College of Management and Information Technology**

**In partial fulfillment of the requirements for the Course**

**java Programming**

**Mid Baneshwor, Kathmandu**

**December 2021**

# Write programto use access control in java.

## Source Code

class Access

{

int a;

public int b;

private int c;

void set(int i)

{

c=i;

}

void show()

{

System.out.println(c);

}

}

class access\_control

{

public static void main(String[] args) {

Access ob=new Access();

ob.a=10;

ob.b=20;

/\*ob.c=30; error\*/

ob.set(30);

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

ob.show();

}

}

## Output Window



# Write program to display average.

## Source Code

import java.util.Scanner;

class average

{

public static void main(String[] args) {

int a,b,c;

Scanner s=new Scanner(System.in);

System.out.print("enter 3 numbers: ");

a=s.nextInt();

b=s.nextInt();

c=s.nextInt();

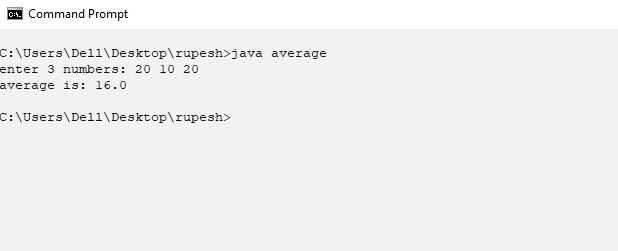
float avg=(a+b+c)/3;

System.out.println("average is: "+avg);

}

}

## Output Window



# Write program to use break.

## Source code

class break\_as\_goto

{

public static void main(String[] args) {

boolean b =true;

first:

{

second:

{

third:

{

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

if(b)

break third;

System.out.println("This won't execute");

}

System.out.println("This won't execute");

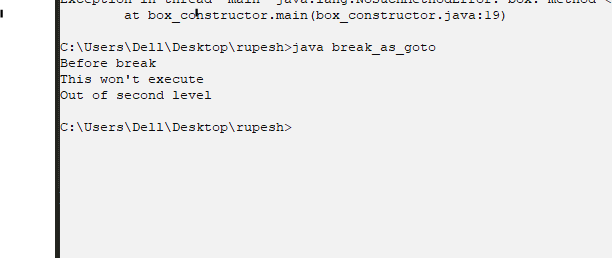
}

System.out.println("Out of second level");

}}

}

## Output window



# Write program to terminate break.

## Source code

class break\_terminateloop

{

public static void main(String[] args) {

int i;

for(i=0;i<100;i++)

{

if(i==10)

break;

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

}

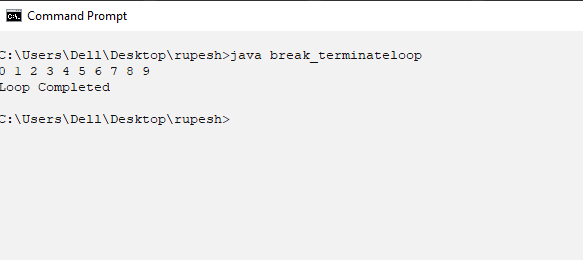
System.out.println();

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

}

}

## Output window



# Write program to call by value.

## Source code

class argument\_pass

{

void set (int i, int j)

{

i\*=5;

j+=5;

}

}

class call\_by\_value

{

public static void main(String[] args) {

argument\_passob = new argument\_pass();

int p = 10; int q = 20;

System.out.println(p+""+q);

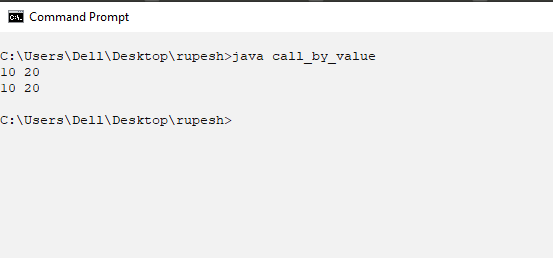
ob.set(p,q);

System.out.println(p+""+q);

}

}

## Output window



# Write a program that display Ambiguity in varargs method.

## Source code

class clg

{

static void display(double... contact)

{

System.out.println("Contacts are: ");

for(double d: contact)

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

System.out.println();

}

static void display(String cname,double... contact)

{

System.out.println("Name: "+cname+"Contacts: ");

for(double d: contact)

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

System.out.println();

}

static void display(boolean... b )

{

System.out.println("Values are: ");

for(boolean x:b)

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

System.out.println();

}

public static void main(String[]args)

{

display();/\*ambiguity occurs\*/

display("KCMIT",8525930,440666);

display("NCC");

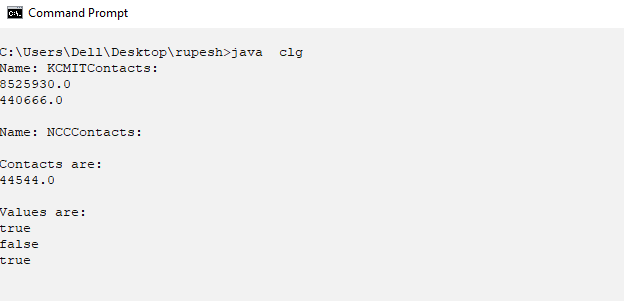
display(44544);

display(true,false,true);

}

}

## Output window



# Write program that take college name and display.

## Source code

class college

{

String name,address;

int rank;

void set(String n, String add, int r)

{

name = n; address = add; rank = r;

}

void show()

{

System.out.println(name+""+address+""+rank);

}

}

class college\_demo

{

public static void main(String[] args) {

college c[] = new college[5];

int i;

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

c[i] = new college();

}

c[0].set("KCMIT","Baneshwor",13);

c[1].set("ABC","Bhaktapur",12);

c[2].set("Orchid","Sinamangal",11);

c[3].set("Global","Baneshwor",15);

c[4].set("Apex","Mid-Baneshwor",14);

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

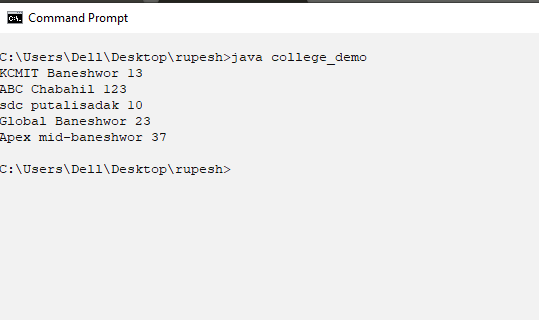
c[i].show();

}

}

}

## Output window



# Write a program that take college name and display using this keyword.

## Source code

class college

{

String name,address;

int rank;

void set(String name, String address, int rank)

{

this.name = name;

this.address = address;

this.rank = rank;

}

void show()

{

System.out.println(name+""+address+""+rank);

}

}

class college\_demothis

{

public static void main(String[] args) {

college c[] = new college[5];

int i;

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

c[i] = new college();

}

c[0].set("KCMIT","Baneshwor",13);

c[1].set("ABC","Bhaktapur",12);

c[2].set("Orchid","Sinamangal",11);

c[3].set("Global","Baneshwor",15);

c[4].set("Apex","Mid-Baneshwor",14);

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

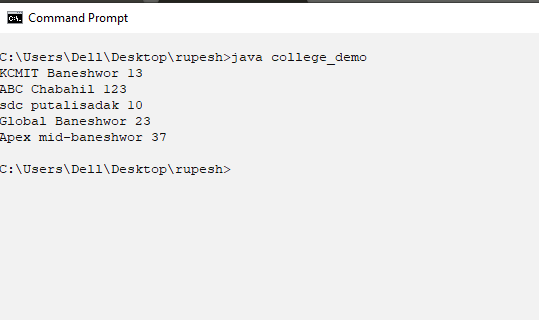
c[i].show();

}

}

}

## Output window



# Write a varargs program.

## Source code

class command\_line

{

public static void main(String[] args) {

int i;

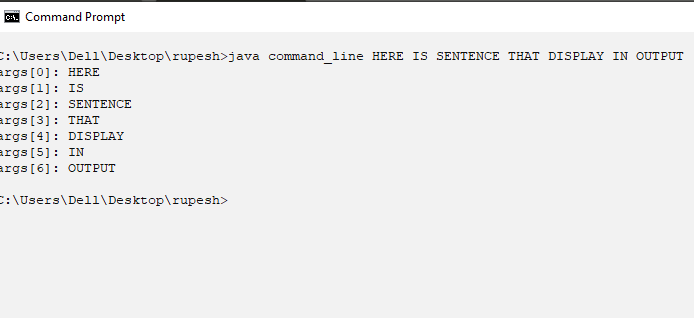
for(i=0;i<args.length;i++)

System.out.println("args["+i+"]: "+args[i]);

}

}

## Output window



# Write a programto use continue in program.

## Source code

class continue\_example

{

public static void main(String[] args) {

int i;

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

{

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

if(i%2==0)

continue;

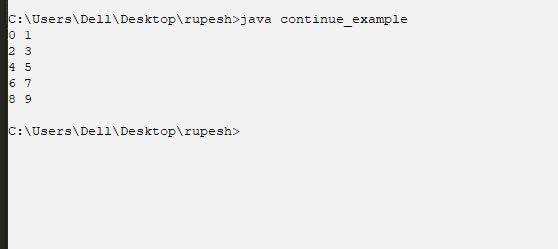
System.out.println();

}

}

}

## Output window



# Write a program of Do while example.

## Source code

class do\_while\_demo

{

public static void main(String[] args) {

int n = 10;

do{

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

n--;

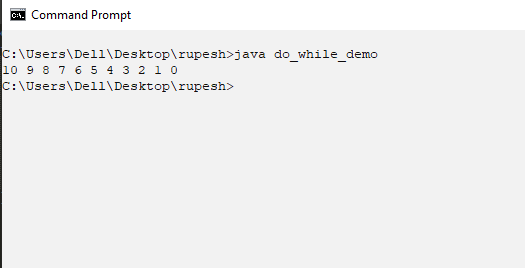
}

while(n>=0);

}

}

## Output window



# Write a program that take doctors details.

## Source code

class doctor{

String specilization;

int reg\_no;

int PAN;

void set\_value(String s,intr,int p)

{

specilization = s;

reg\_no = r;

PAN = p;

}

void display()

{

System.out.println("specilization is: "+specilization+

"reg\_no is: " +reg\_no+ "PAN"+PAN);

}

}

class method\_parameter

{

public static void main(String[] args) {

doctor d=new doctor();

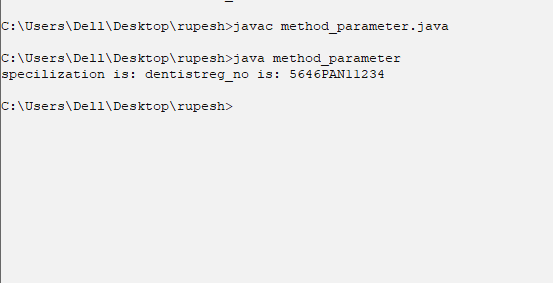
d.set\_value("dentist",5646,11234);

d.display();

}

}

## Output window



# Write a factorial program .

## Source code

class factorial

{

int fact(int n)

{

if(n==1)

return 1;

else

return fact(n-1)\*n;

}

}

class factorial\_example

{

public static void main(String[] args)

{

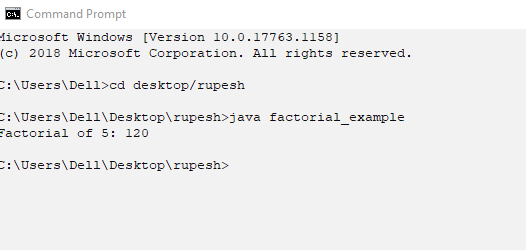
factorial ob = new factorial();

System.out.println("Factorial of 5: "+ob.fact(5));

}

}

## Output window



# Write a program that increment the given variable.

## Source code

class incrementexample

{

public static void main(String[] args) {

int a=1,b=2,c,d;

c=++b;

d=a++;

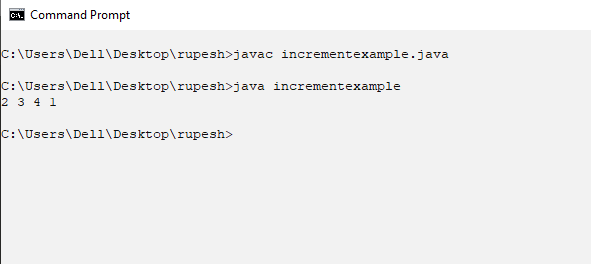
c++;

System.out.println (a+""+b+""+c+""+d);

}

}

## Output window



# Write a program of nested-class.

## Source code

class outer

{

int a = 50;

class inner

{

void show()

{

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

}

}

void display()

{

inner ob = new inner();

ob.show();

}

}

class inner\_example

{

public static void main(String[]args){

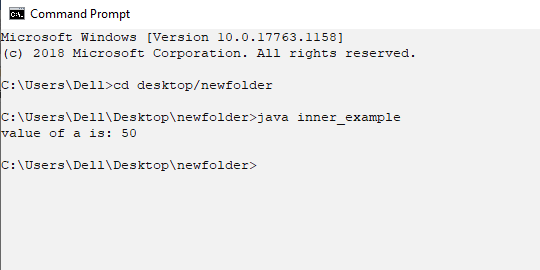
outer out = new outer();

out.display();

}

}

## Output window



# Write program to return the Boolean .

## Source code

class return\_demo

{

public static void main(String[] args) {

boolean b=true;

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

if(b)

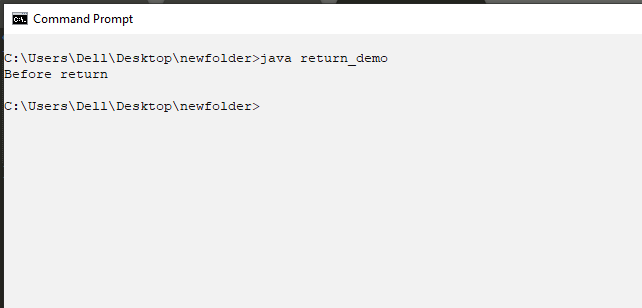
return;

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

}

}

## Output window



# Write a program that display the length of room less than 15

## Source code

class room

{

int length,breath,height;

void set(int l,int b, int h)

{

length=l; breath=b;height=h;

}

void show()

{

System.out.println("volume is: "+length\*breath\*height);

}

}

class room\_demo

{

public static void main(String[] args) {

room r[]=new room[7];

int i;

for(i=0;i<7;i++)

{

r[i]=new room();

}

r[0].set(20,10,5);

r[1].set(10,8,5);

r[2].set(18,15,10);

r[3].set(5,3,4);

r[4].set(16,11,6);

r[5].set(19,17,3);

r[6].set(12,8,4);

for(i=0;i<7;i++)

{

if (r[i].length<15)

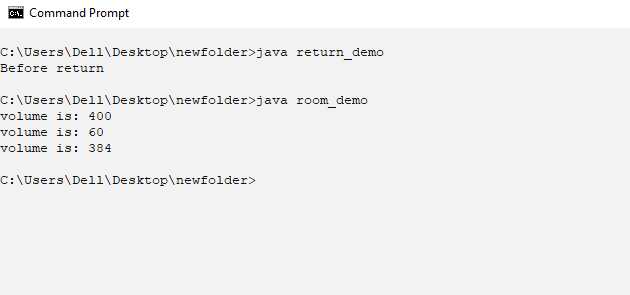
r[i].show();

}

}

}

## Output window



# Write a program of static keyword .

## Source code

/\*static keyword\*/

class static\_demo

{

static int a = 100;

static int b;

static void show(int c)

{

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

}

static

{

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

b= a+50;

}

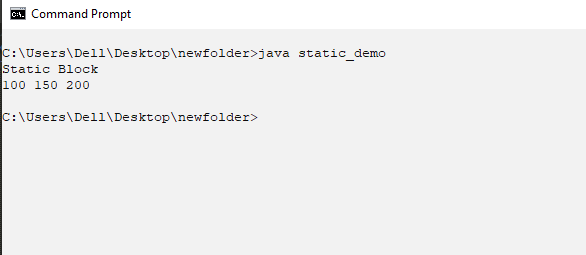
public static void main(String[]args){

show(200);

}

}

## Output window



# Write program that display the switch program.

## Source code

import java.util.Scanner;

class user\_input

{

public static void main(String[] args) {

String name; int age,roll;

float height; char gender;

Scanner s = new Scanner(System.in);

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

name = s.nextLine();

System.out.println("enter age and roll: ");

age = s.nextInt();

roll = s.nextInt();

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

height = s.nextFloat();

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

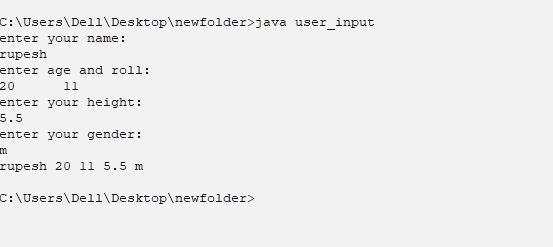
gender = s.next().charAt(0);

System.out.println(name+""+age+""+roll+""+height+""+gender);

}

}

## Output window



# Write a program that display default constructor.

## Source code

class rectangle

{

int length,breadth;

rectangle()/\*this is default constructor\*/

{

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

}

rectangle(int length, int breadth)/\*this is parameterized constructor\*/

{

this.length = length;

this.breadth = breadth;

System.out.println("Area: "+length\*breadth);

}

}

class xyz

{

public static void main(String[]args)

{

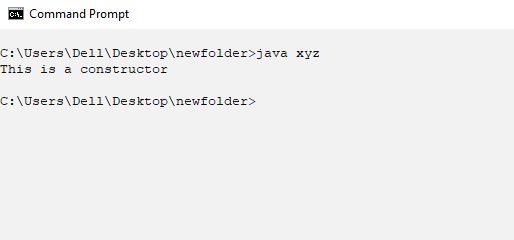
rectangle ob1 = new rectangle(15,9);/\*this is parameterized constructor\*/

rectangle ob2 = new rectangle();/\*this is default constructor\*/

}

}

## Output window



# Write program display name and salary by using balance(string ,int).

## Source code

class balance

{

String name;

int bal;

balance(String n, int b)

{

name = n;

bal = b;

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

}

}

class xyz\_demo

{

public static void main(String[] args)

{

balance ob[] = new balance[3];

ob[0] = new balance("Ram",12345);

ob[1] = new balance("Shyam",20000);

ob[2] = new balance("Rupesh",15000);

}

}

## Output window



# Write program to check a number is even or odd.

## Source code

class vowel

{

public static void main(String [] args) {

char a='a';

if(a=='a'||a=='e'||a=='i'||a=='o'||a=='u'||a=='A'||a=='E'||a=='I'||a=='O'||a=='U')

System.out.println("it is vowel ");

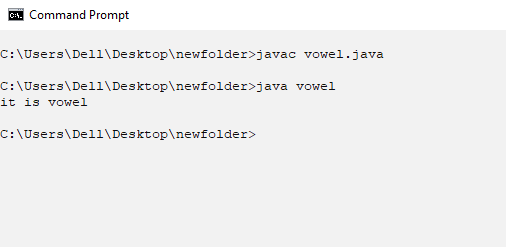
else

System.out.println("it is not vowel ");

}

}

## Output window



# Write a program of Two D array

## Source code

class twodarray

{

public static void main(String[] args) {

int arr[][] = new int[4][4];

int i,j,k=1;

for(i=0;i<4;i++)

for (j=0;j<i+1 ;j++ ) {

arr[i][j]=k;

k=k+2;

}

for(i=0;i<4;i++){

for (j=0;j<i+1 ;j++ ) {

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

System.out.println();

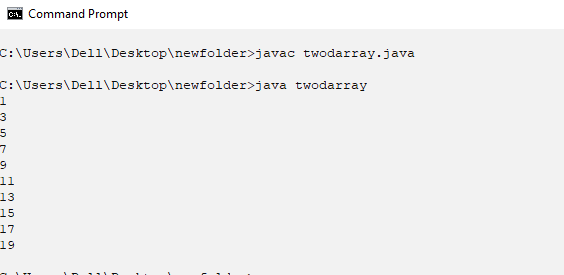
}

}

}

}

## Output window



# Write super keyword to call constructor.

## Source code

class base

{

int length,breadth;

base (int l, int b)

{

length=l;

breadth=b;

System.out.println("Area of room:"+(length\*breadth));

}

}

class derived extends base

{

int height;

derived (int l,int b)

{

super(l,b);

}

derived (int l,intb,int h)

{

super(l,b);

height=h;

System.out.println("heigth of room:"+height);

System.out.println("volume of room:"+(length\*breadth\*height));

}

}

class constructor\_demo

{

public static void main(String []args)

{

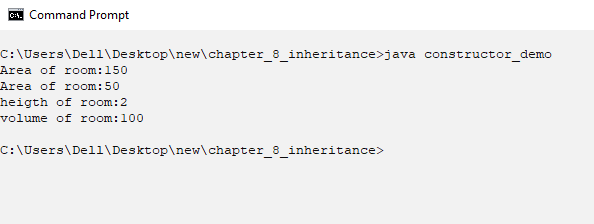
derived ob1=new derived (10,15);

derived ob=new derived (10,5,2);

}

}

## Output window



# Write dynamic\_polymorphism.

## Source code

class A

{

void abc()

{

System.out.println("kcmit");

}

}

class B extends A

{

void abc()

{

System.out.println("1234");

}

}

class dynamic\_polymorphism

{

public static void main(String []args)

{

A a=new A();

B b=new B();

A ref;

ref=a;

ref.abc();

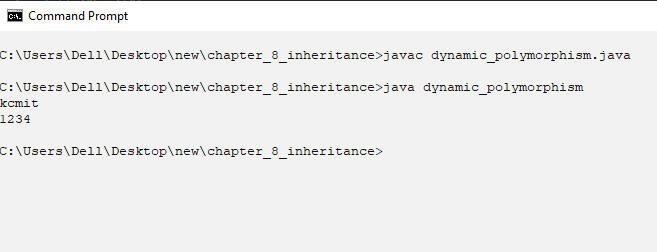
ref=b;

ref.abc();

}

}

## Output window



# Write method overriding

## Source code

class A

{

void display()

{

System.out.println("it is super class method");

}

}

class B extends A

{

void display()//override the display method of super class

{

System.out.println("it is sub class method");

}

}

class method\_overriding

{

public static void main(String []args)

{

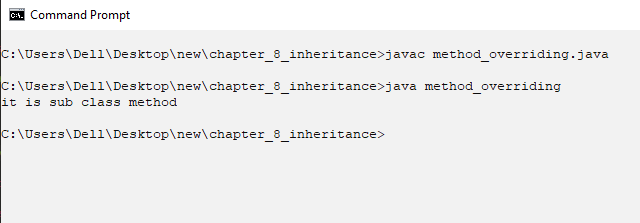
B a= new B();

a.display();

}

}

## Output window



# Write one superclass with multiple subclass

## Source code

//one superclass multiple subclass

class one

{

int length;

one(int l)

{

length=l;

show1();

}

void show1()

{

System.out.println("length of room:"+length);

}

}

class two extends one

{

int breadth;

two(int l,int b)

{

super(l);//one(int l)

breadth=b;

show2();

}

void show2()

{

System.out.println("breadth of room:"+breadth);

}

}

class three extends two

{

int height;

three (int l,intb,int h)

{

super(l,b);//call two(int l,int b)

height=h;

show3();

System.out.println("volume of box:"+(length\*breadth\*height));

}

void show3()

{

System.out.println("height of box:"+height);

}

}

class multi\_constructor

{

public static void main(String []args)

{

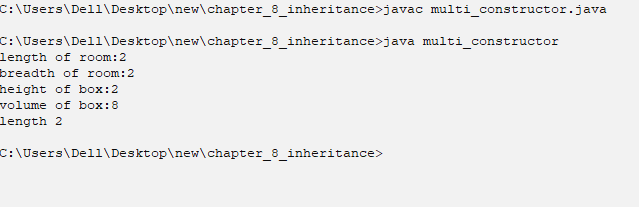
three ob=new three(2,2,2);

System.out.println("length "+(ob.length));

}

}

## Output window



# Write an example of runtime polymorphism

## Source code

class info

{

String name;int age;

info(String s, int a)

{

name=s;

age=a;

}

void display()

{

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

}

}

class overriding extends info

{

int phone;

overriding(String s,inta,intph)

{

super(s,a);

phone=ph;

}

void display()//method overridden

{

System.out.println("name: "+name+" age:"+age+" phone no:"+phone);

}

}

class runtime\_polymorphism

{

public static void main(String []args)

{

info ob1=new info("Ram",20);

overriding ob=new overriding("Aashish",19,222222);

info ref;

ref=ob1;//upcasting

ref.display();//print the method of superclass:info as ob1 is object of superclass

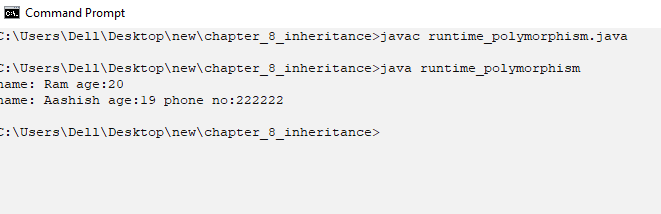
ref=ob;//upcasting

ref.display();//print the method of subclass only as ob is object of subclass:overriding

}

}

## Output window



# Write super keyword on method overriding.

## Source code

//use of super keyword on method overriding

class info

{

String name;int age;

info(String s, int a)

{

name=s;

age=a;

}

void display()

{

System.out.print("name: "+name+" age:"+age);

}

}

class overriding extends info

{

int phone;

overriding(String s,inta,intph)

{

super(s,a);

phone=ph;

}

void display()//method overridden

{

super.display();//super keyword is used to call super class method

System.out.println(" phone no:"+phone);

}

}

class super\_method\_overriding

{

public static void main(String []args)

{

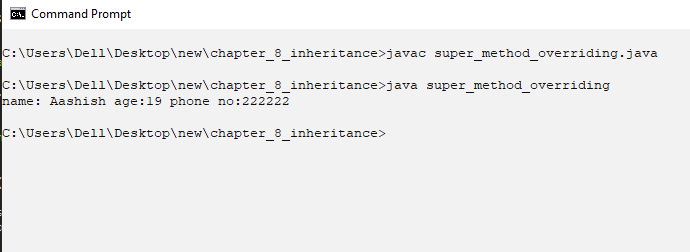
overriding ob=new overriding("Aashish",19,222222);

ob.display();

}

}

## Output window



# Write arthematic\_try -catch.

## Source code

class arthematic\_trycatch

{

public static void main(String []args)

{

System.out.println("before");

try{

int a=10/0;

System.out.println(+a);

}catch(Exception e){

System.out.println("error found :"+e);

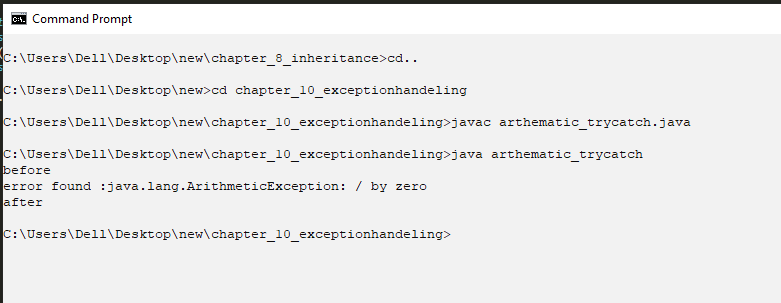
}

System.out.println("after");//exception is handeled so this line will also be print

}

}

## Output window



# Write multiple exception with multiple catch

## Source code

class multi\_exception{

public static void main(String []args){

System.out.println("before");

try{

int b[]=new int[3];

b[10]=9;

int a=5/0;

}

catch (ArithmeticExceptiona){

System.out.println(a);

}

catch(ArrayIndexOutOfBoundsException e){

System.out.println(e);

}

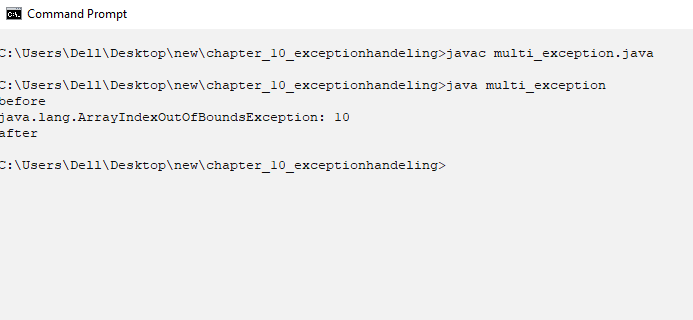
System.out.println("after");

}

}

//only ArrayIndexOutOfBoundsException will be handeled as it is in first in try

## Output window



# Write a program to show nested try-catch.

## Source code

//nested try

class nested\_try

{

public static void main(String []args){

System.out.println("before");

try{

try{

int b[]=new int[3];

b[9]=8;

}catch(Exception e){

System.out.println("exception caught"+e);

}

int a=5/0;

}catch(ArithmeticException a){

System.out.println("exception caught"+a);

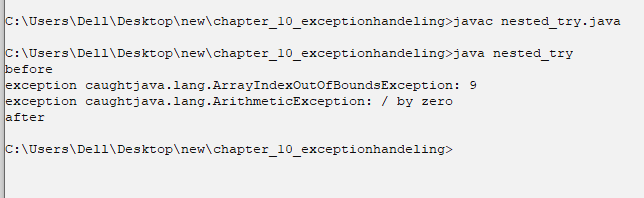
}

System.out.println("after");

}

}

## Output window



# Write throw exception.

## Source code

//throw exception

import java.util.Scanner;

class A{

int b;

void demo(int a)

{

b=a;

if (b<18){

throw new ArithmeticException("error");

}

else{

System.out.println("hello");

}

}

}

class throw\_demo{

public static void main(String []args){

Scanner s=new Scanner(System.in);

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

int age=s.nextInt();

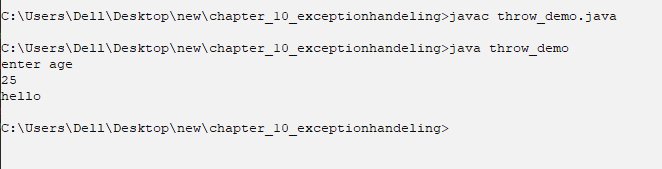
A ob=new A();

ob.demo(age);

}

}

## Output window



# Take a input from user and make Input output file

## Source code

import java.io.\*;

import java.util.Scanner;

class fill

{

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

{

try

{

FileWriterfw=new FileWriter("Emp.txt");

BufferedWriterbw= new BufferedWriter(fw);

String name,id,DOB;

Scanner scan=new Scanner(System.in);

System.out.print("Enter name,id and Dob: ");

name=scan.nextLine();

id=scan.nextLine();

DOB=scan.nextLine();

bw.write(id);

bw.write(DOB);

bw.write(name);

bw.close();

}

catch(Exception e)

{

System.out.println(e);

}

try

{

FileReaderfr = new FileReader("Emp.txt");

BufferedReaderbr = new BufferedReader(fr);

String msg=null;

while((msg=br.readLine())!=null)

{

System.out.println(msg);

}

br.close();

}catch(Exception e)

{

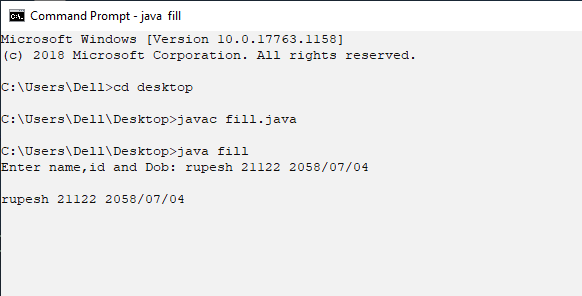
System.out.println(e);

}

}

}

## Output window



Contents

[1 Write programto use access control in java. 1](#_Toc64218654)

[1.1 Source Code 1](#_Toc64218655)

[1.2 Output Window 2](#_Toc64218656)

[2 Write program to display average. 3](#_Toc64218657)

[2.1 Source Code 3](#_Toc64218658)

[2.2 Output Window 3](#_Toc64218659)

[3 Write program to use break. 4](#_Toc64218660)

[3.1 Source code 4](#_Toc64218661)

[3.2 Output window 4](#_Toc64218662)

[4 Write program to terminate break. 5](#_Toc64218663)

[4.1 Source code 5](#_Toc64218664)

[4.2 Output window 5](#_Toc64218665)

[5 Write program to call by value. 6](#_Toc64218666)

[5.1 Source code 6](#_Toc64218667)

[5.2 Output window 6](#_Toc64218668)

[6 Write a program that display Ambiguity in varargs method. 7](#_Toc64218669)

[6.1 Source code 7](#_Toc64218670)

[6.2 Output window 8](#_Toc64218671)

[7 Write program that take college name and display. 9](#_Toc64218672)

[7.1 Source code 9](#_Toc64218673)

[7.2 Output window 10](#_Toc64218674)

[8 Write a program that take college name and display using this keyword. 11](#_Toc64218675)

[8.1 Source code 11](#_Toc64218676)

[8.2 Output window 12](#_Toc64218677)

[9 Write a varargs program. 13](#_Toc64218678)

[9.1 Source code 13](#_Toc64218679)

[9.2 Output window 13](#_Toc64218680)

[10 Write a programto use continue in program. 14](#_Toc64218681)

[10.1 Source code 14](#_Toc64218682)

[10.2 Output window 14](#_Toc64218683)

[11 Write a program of Do while example. 15](#_Toc64218684)

[11.1 Source code 15](#_Toc64218685)

[11.2 Output window 15](#_Toc64218686)

[12 Write a program that take doctors details. 16](#_Toc64218687)

[12.1 Source code 16](#_Toc64218688)

[12.2 Output window 17](#_Toc64218689)

[13 Write a factorial program . 18](#_Toc64218690)

[13.1 Source code 18](#_Toc64218691)

[13.2 Output window 18](#_Toc64218692)

[14 Write a program that increment the given variable. 19](#_Toc64218693)

[14.1 Source code 19](#_Toc64218694)

[14.2 Output window 19](#_Toc64218695)

[15 Write a program of nested-class. 20](#_Toc64218696)

[15.1 Source code 20](#_Toc64218697)

[15.2 Output window 21](#_Toc64218698)

[16 Write program to return the Boolean . 22](#_Toc64218699)

[16.1 Source code 22](#_Toc64218700)

[16.2 Output window 22](#_Toc64218701)

[17 Write a program that display the length of room less than 15 23](#_Toc64218702)

[17.1 Source code 23](#_Toc64218703)

[17.2 Output window 24](#_Toc64218704)

[18 Write a program of static keyword . 25](#_Toc64218705)

[18.1 Source code 25](#_Toc64218706)

[18.2 Output window 25](#_Toc64218707)

[19 Write program that display the switch program. 26](#_Toc64218708)

[19.1 Source code 26](#_Toc64218709)

[19.2 Output window 27](#_Toc64218710)

[20 Write a program that display default constructor. 28](#_Toc64218711)

[20.1 Source code 28](#_Toc64218712)

[20.2 Output window 29](#_Toc64218713)

[21 Write program display name and salary by using balance(string ,int). 30](#_Toc64218714)

[21.1 Source code 30](#_Toc64218715)

[21.2 Output window 31](#_Toc64218716)

[22 Write program to check a number is even or odd. 32](#_Toc64218717)

[22.1 Source code 32](#_Toc64218718)

[22.2 Output window 32](#_Toc64218719)

[23 Write a program of Two D array 33](#_Toc64218720)

[23.1 Source code 33](#_Toc64218721)

[23.2 Output window 33](#_Toc64218722)

[24 Write super keyword to call constructor. 34](#_Toc64218723)

[24.1 Source code 34](#_Toc64218724)

[24.2 Output window 35](#_Toc64218725)

[25 Write dynamic\_polymorphism. 36](#_Toc64218726)

[25.1 Source code 36](#_Toc64218727)

[25.2 Output window 37](#_Toc64218728)

[26 Write method overriding 38](#_Toc64218729)

[26.1 Source code 38](#_Toc64218730)

[26.2 Output window 38](#_Toc64218731)

[27 Write one superclass with multiple subclass 39](#_Toc64218732)

[27.1 Source code 39](#_Toc64218733)

[27.2 Output window 40](#_Toc64218734)

[28 Write an example of runtime polymorphism 41](#_Toc64218735)

[28.1 Source code 41](#_Toc64218736)

[28.2 Output window 42](#_Toc64218737)

[29 Write super keyword on method overriding. 43](#_Toc64218738)

[29.1 Source code 43](#_Toc64218739)

[29.2 Output window 44](#_Toc64218740)

[30 Write arthematic\_try -catch. 45](#_Toc64218741)

[30.1 Source code 45](#_Toc64218742)

[30.2 Output window 45](#_Toc64218743)

[31 Write multiple exception with multiple catch 46](#_Toc64218744)

[31.1 Source code 46](#_Toc64218745)

[31.2 Output window 46](#_Toc64218746)

[32 Write a program to show nested try-catch. 47](#_Toc64218747)

[32.1 Source code 47](#_Toc64218748)

[32.2 Output window 47](#_Toc64218749)

[33 Write throw exception. 48](#_Toc64218750)

[33.1 Source code 48](#_Toc64218751)

[33.2 Output window 48](#_Toc64218752)

[34 Take a input from user and make Input output file 49](#_Toc64218753)

[34.1 Source code 49](#_Toc64218754)

[34.2 Output window 50](#_Toc64218755)