

**LAB PROJECT**

**Course Name:**

* Software Engineering

**Topic:**

* Lab Project (42 Lab Tasks)

**Submitted By:**

* Farooq Ali (12391)
* Altahash Butt (12366)

**Submitted To:**

* Ma’am Sadaf Anwar

**Section:**

* BSSE 2B (Morning)

**NATIONAL UNIVERSITY OF MODERN LANGUAGES**

**CONTENTS**

[ Lab Task 1: First oop Lab 3](#_Toc71119463)

[ Lab Task 2: 3](#_Toc71119464)

[ Lab Task 3: 4](#_Toc71119465)

[ Lab Task 4: 5](#_Toc71119466)

[ Lab Task 5: 6](#_Toc71119467)

[ Lab Task 6: 7](#_Toc71119468)

[ Lab Task 7: 8](#_Toc71119469)

[ Lab Task 8: 9](#_Toc71119470)

[ Lab Task 9: 10](#_Toc71119471)

[ Lab Task 10: 11](#_Toc71119472)

[ Lab Task 11: 12](#_Toc71119473)

[ Lab Task 12: 13](#_Toc71119474)

[ Lab Task 13: BOX 14](#_Toc71119475)

[ Lab Task 14: PASS BY VALUE 16](#_Toc71119476)

[ Lab Task 15: PASS BY REFERENCE 17](#_Toc71119477)

[ lab Task 16: Passing Object as argument 18](#_Toc71119478)

[ lab Task 17: Passing Object in constructor 20](#_Toc71119479)

[ lab Task 18: Returning Objects 21](#_Toc71119480)

[ lab Task 19: INHERITANCE 22](#_Toc71119481)

[ lab Task 20: inheritance Box Example 24](#_Toc71119482)

[ lab Task 21: inheritance (BOX + WEIGht): 25](#_Toc71119483)

[ Lab Task 22: Inheritance (Box model + Weight + Shipping): 27](#_Toc71119484)

[ Lab Task 23: use of THIS keyword 31](#_Toc71119485)

[ lab Task 24 : Static Example 32](#_Toc71119486)

[ lab Task 25 : Static Example (variables + methods): 33](#_Toc71119487)

[ lab Task 26 : ENCAPSULATED STUDENTS 34](#_Toc71119488)

[ lab Task 27 : Encapsulation 36](#_Toc71119489)

[ lab Task 28 : Abstract 37](#_Toc71119490)

[ lab Task 29: Method Overloading 39](#_Toc71119491)

[ lab Task 30 : Static and Instance Blocks 40](#_Toc71119492)

[ lab Task 31: Final Keyword 41](#_Toc71119493)

[ lab Task 32: Dynamic Method Dispatch 42](#_Toc71119494)

[ lab Task 33: Car Details Using Scanner 44](#_Toc71119495)

[ lab Task 34: Car Example Using Constructor 46](#_Toc71119496)

[ Lab Task 35: CIRCLE EXAMPLE 48](#_Toc71119497)

[ Lab task 36: Savings Account 49](#_Toc71119498)

[ Lab Task 37: TIME TASK 51](#_Toc71119499)

[ LAB TASK 38: CLOCK Task 53](#_Toc71119500)

[ Lab task 39: Abstract class and Method 54](#_Toc71119501)

[ Lab Task 40: Employee Task 56](#_Toc71119502)

[ Lab Task 41: Task Game 59](#_Toc71119503)

[ Lab Task 42: Interface Month Numbers 60](#_Toc71119504)

[ Lab Task 43: Interface Register For Exams 61](#_Toc71119505)

# Lab Task 1: First oop Lab

**CODE:**

package lab1task1;

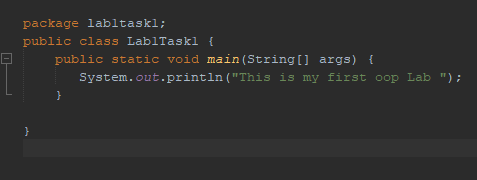
public class Lab1Task1 {

public static void main(String[] args) {

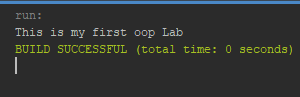
System.out.println("This is my first oop Lab ");

}

}



**OUTPUT:**

****

# Lab Task 2:

**CODE:**

package lab1task2;

public class Lab1Task2 {

public static void main(String[] args) {

int j,k,p,q,r,s,t;

j=5;

k=2;

p=j+k;

q=j-k;

r=j\*k;

s=j/k;

t=j%k;

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

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

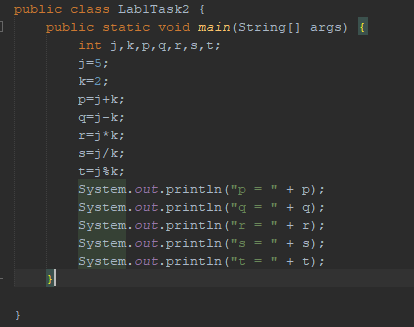
System.out.println("r = " + r);

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

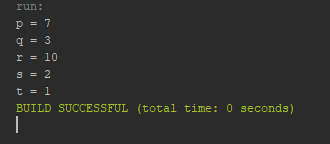
System.out.println("t = " + t);

}

}



**OUTPUT:**

****

# Lab Task 3:

**CODE:**

package lab1task3;

public class Lab1Task3 {

public static void main(String[] args) {

int j, p, q, r, s;

j=5;

p=++j;

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

q=j++;

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

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

r=--j;

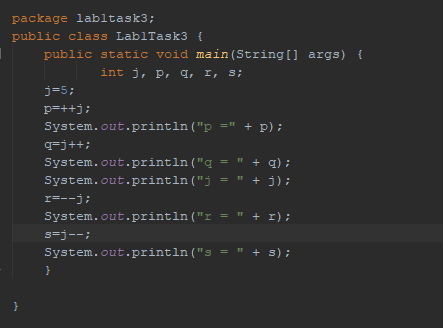
System.out.println("r = " + r);

s=j--;

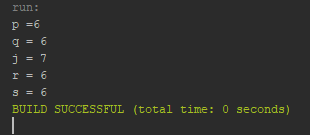
System.out.println("s = " + s);

}

}



**OUTPUT:**

****

# Lab Task 4:

**CODE:**

package lab1task4;

public class Lab1Task4 {

public static void main(String[] args) {

int p=2;

int q=2;

int r=3;

System.out.println("p<r "+(p<r));

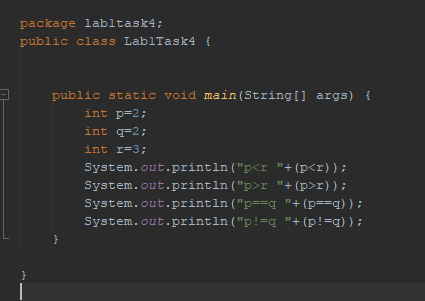
System.out.println("p>r "+(p>r));

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

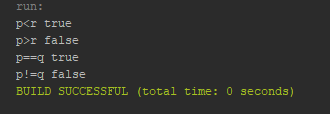
System.out.println("p!=q "+(p!=q));

}

}



**OUTPUT:**

****

# Lab Task 5:

**CODE:**

package lab1task5;

public class Lab1Task5 {

public static void main(String[] args) {

boolean t=true;

boolean f=false;

System.out.println("f && f "+(f && f));

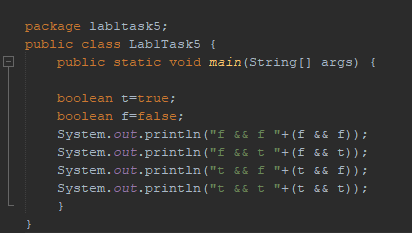
System.out.println("f && t "+(f && t));

System.out.println("t && f "+(t && f));

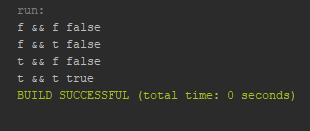
System.out.println("t && t "+(t && t));

}

}



**OUTPUT:**

****

# Lab Task 6:

**CODE:**

package lab1task6;

public class Lab1Task6 {

public static void main(String[] args) {

boolean t = true;

boolean f = false;

System.out.println("f || f " + (f || f));

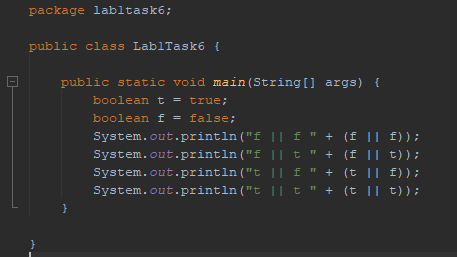
System.out.println("f || t " + (f || t));

System.out.println("t || f " + (t || f));

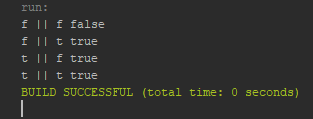
System.out.println("t || t " + (t || t));

}

}



**OUTPUT:**

****

# Lab Task 7:

**CODE:**

package lab1task7;

public class Lab1Task7 {

public static void main(String[] args) {

boolean b;

int j, k;

j = 0;

k = 0;

b = (j++ == k) && (j == ++k);

System.out.println("b,j,k " + b + "," + j + "," + k);

j = 0;

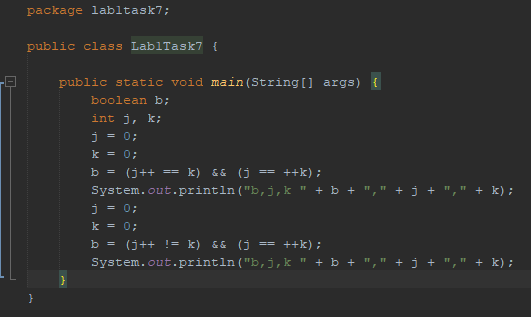
k = 0;

b = (j++ != k) && (j == ++k);

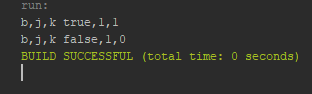
System.out.println("b,j,k " + b + "," + j + "," + k);

}

}



**OUTPUT:**

****

# Lab Task 8:

**CODE:**

package lab1task8;

public class Lab1Task8 {

public static void main(String[] args) {

int fact = 1;

int num = 4; //Factorial of num 4 will be taken

for (int i = num; i > 1; i--) {

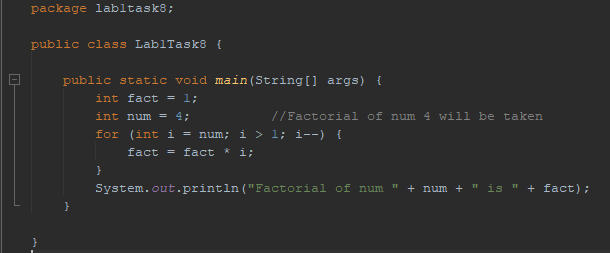
fact = fact \* i;

}

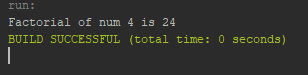
System.out.println("Factorial of num " + num + " is " + fact);

}

}



**OUTPUT:**



# Lab Task 9:

**CODE:**

package lab1task9;

public class Lab1Task9 {

public static void main(String[] args) {

int rows = 9;

for (int i = rows; i > 0; i--) {

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

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

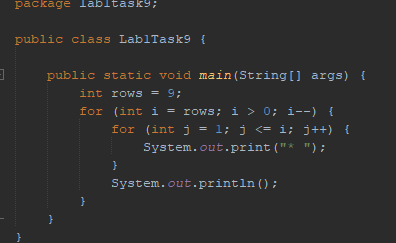
}

System.out.println();

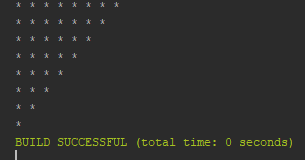
}

}

}



**OUTPUT:**

****

# Lab Task 10:

**CODE:**

package lab1task10;

public class Lab1Task10 {

public static void main(String[] args) {

boolean T = true;

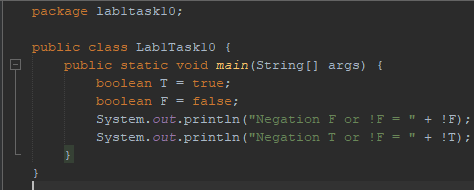
boolean F = false;

System.out.println("Negation F or !F = " + !F);

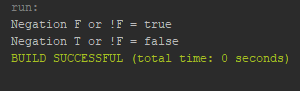
System.out.println("Negation T or !F = " + !T);

}

}



**OUTPUT:**

****

# Lab Task 11:

**CODE:**

package lab1task11;

public class Lab1Task11 {

public static void main(String[] args) {

int L = 10;

int M = 12;

int N = 25;

System.out.println("L<=M " + (L <= M));

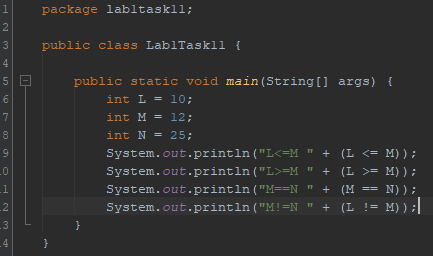
System.out.println("L>=M " + (L >= M));

System.out.println("M==N " + (M == N));

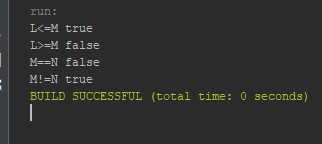
System.out.println("M!=N " + (L != M));

}

}



**OUTPUT:**

****

# Lab Task 12:

**CODE:**

package lab1task12;

public class Lab1Task12 {

public static void main(String[] args) {

int x = 8;

int y = x < 0 ? -x : x;

System.out.println("ABS value of " + x + " is " + y);

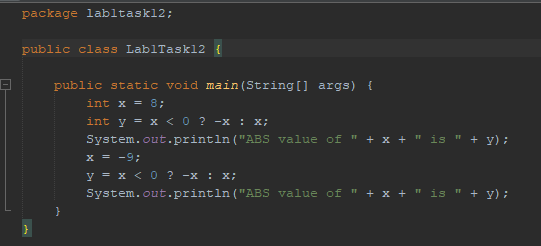
x = -9;

y = x < 0 ? -x : x;

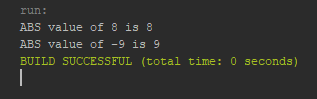
System.out.println("ABS value of " + x + " is " + y);

}

}



**OUTPUT:**

****

# Lab Task 13: BOX

**CODE:**

package labtask13;

class box {

double width;

double height;

double depth;

void setDim(double w, double h, double d) {

width = w;

height = h;

depth = d;

}

double volume() {

return width \* height \* depth;

}

}

public class LabTask13 {

public static void main(String[] args) {

box mybox1 = new box();

box mybox2 = new box();

double vol;

mybox1.setDim(15, 40, 45);

mybox2.setDim(2, 5, 7);

vol = mybox1.volume();

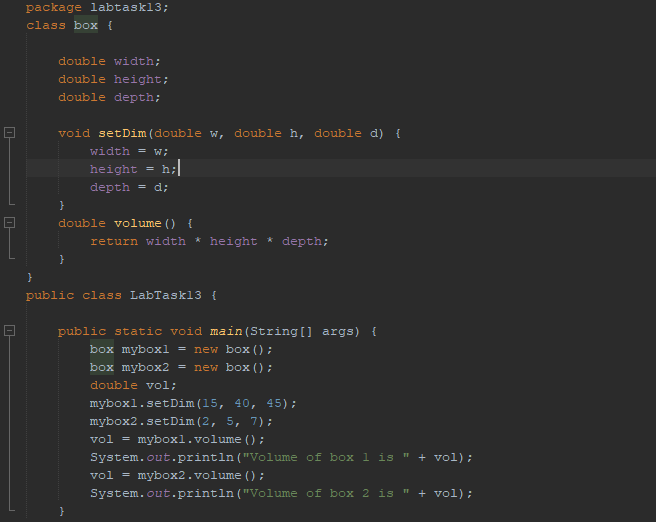
System.out.println("Volume of box 1 is " + vol);

vol = mybox2.volume();

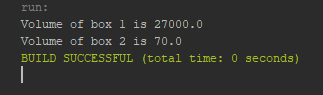
System.out.println("Volume of box 2 is " + vol);

}

}



**OUTPUT:**

****

# Lab Task 14: PASS BY VALUE

**CODE:**

package labtask14;

class passbyvalue {

int a, b;

passbyvalue(int i, int j) {

a = i;

b = j;

}

void change(int i, int j) {

i \*= 2;

j \*= 2;

}

}

public class LabTask14 {

public static void main(String[] args) {

passbyvalue obj = new passbyvalue(1, 2);

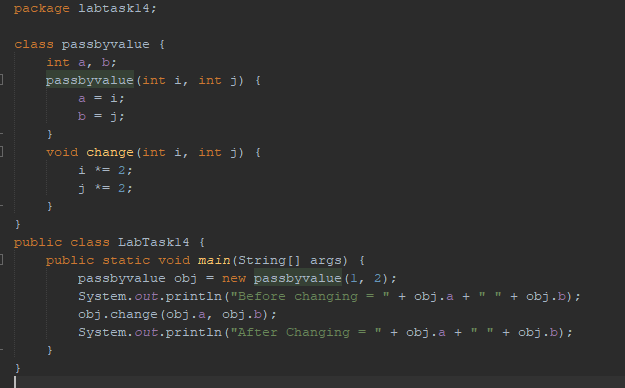
System.out.println("Before changing = " + obj.a + " " + obj.b);

obj.change(obj.a, obj.b);

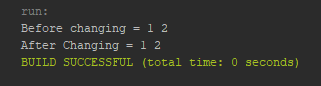
System.out.println("After Changing = " + obj.a + " " + obj.b);

}

}



**OUTPUT:**

****

# Lab Task 15: PASS BY REFERENCE

**CODE:**

package labtask15;

class test{

int a,b;

void passbyReference(test obj){

obj.a\*=2;

obj.b\*=3;

}

}

public class LabTask15 {

public static void main(String[] args){

test ob1=new test();

ob1.a=2;

ob1.b=3;

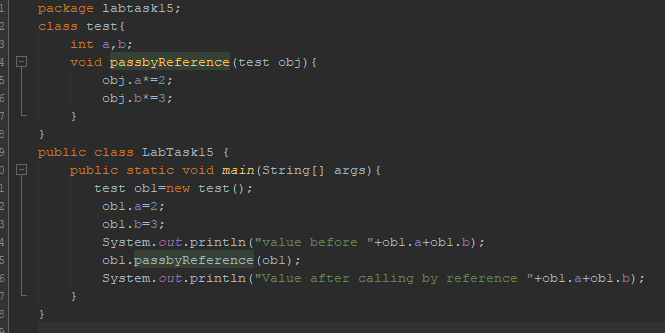
System.out.println("value before "+ob1.a+ob1.b);

ob1.passbyReference(ob1);

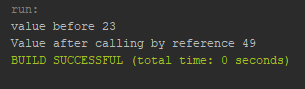
System.out.println("Value after calling by reference "+ob1.a+ob1.b);

}

}



**OUTPUT:**

****

# lab Task 16: Passing Object as argument

**CODE:**

package labtask16;

class objectpass {

int a, b;

objectpass(int i, int j) {

a = i;

b = j;

}

boolean equal(objectpass temp) {

if (temp.a == a && temp.b == b) {

return true;

} else {

return false;

}

}

}

public class LabTask16 {

public static void main(String[] args) {

objectpass ob1=new objectpass(1,2);

objectpass ob2=new objectpass(1,2);

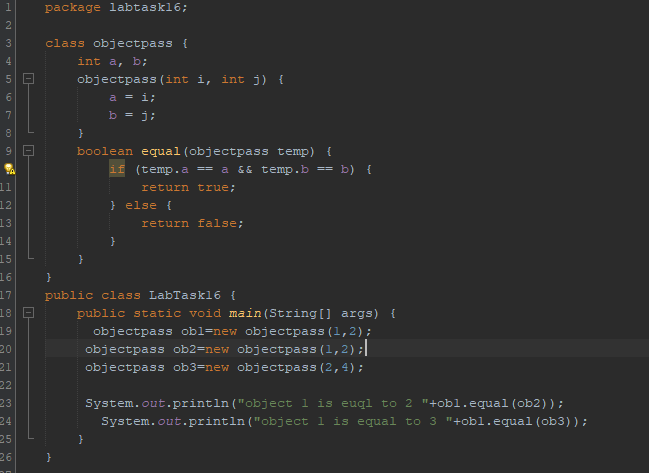
objectpass ob3=new objectpass(2,4);

System.out.println("object 1 is euql to 2 "+ob1.equal(ob2));

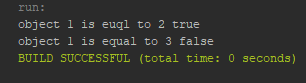
System.out.println("object 1 is equal to 3 "+ob1.equal(ob3));

}

}



**OUTPUT:**

****

# lab Task 17: Passing Object in constructor

**CODE:**

package labtask17;

class box{

int height,width,depth;

box(int i,int j,int k){

height=i;width=j;depth=k;

}

box(box o){

height=o.height;depth=o.depth;width=o.width;

}

int volume(){

return width\*height\*depth;

}

}

public class LabTask17 {

public static void main(String[] args) {

box b1=new box(1,2,3);

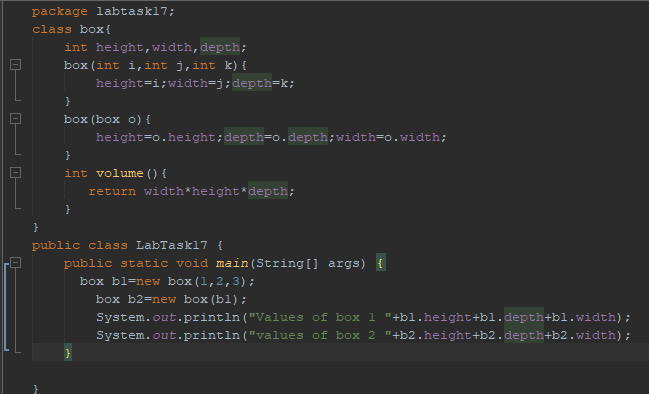
box b2=new box(b1);

System.out.println("Values of box 1 "+b1.height+b1.depth+b1.width);

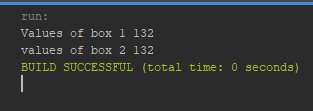
System.out.println("values of box 2 "+b2.height+b2.depth+b2.width);

}

}



**OUTPUT:**

****

# lab Task 18: Returning Objects

**CODE:**

package labtask18;

class test{

int a;

test(int i){

a=i;

}

test incrbyten(){

test temp=new test(a+10);

return temp;

}

}

public class LabTask18 {

public static void main(String[] args) {

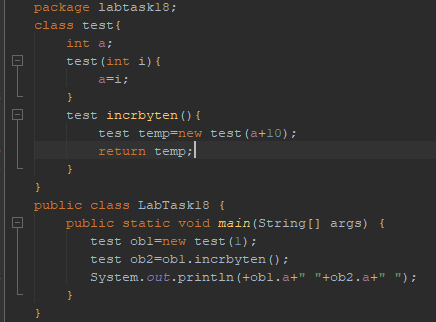
test ob1=new test(1);

test ob2=ob1.incrbyten();

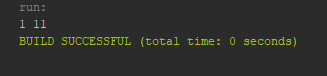
System.out.println(+ob1.a+" "+ob2.a+" ");

}

}

****

**OUTPUT:**

****

# lab Task 19: INHERITANCE

**CODE:**

package labtask19;

class A {

int i;

void showi() {

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

}

}

class B extends A {

int j;

void showj() {

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

}

void sum() {

System.out.println("i+j: " + (i + j));

}

}

public class LabTask19 {

public static void main(String[] args) {

A a=new A();

B b=new B();

System.out.println("Contents of a: ");

a.showi();

b.i=7; b.j=8;

System.out.println("Contents of b: ");

b.showi();

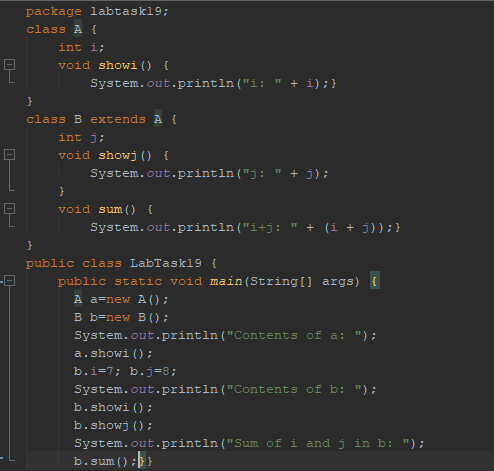
b.showj();

System.out.println("Sum of i and j in b: ");

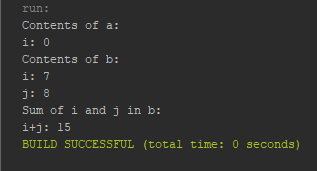
b.sum();

}

}



**OUTPUT:**

****

# lab Task 20: inheritance Box Example

**CODE:**

package labtask20;

class box{

int length,breadth;

box(int l, int b){

length=l;

breadth=b;

}

int area(){

return length\*breadth;

}

}

class box3d extends box{

int height;

box3d(int l,int b,int h){

super(l,b);height=h;

}

int volume(){

return length\*breadth\*height; }

}

public class LabTask20 {

public static void main(String[] args) {

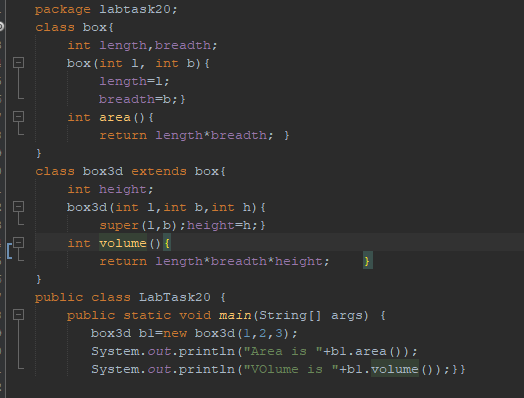
box3d b1=new box3d(1,2,3);

System.out.println("Area is "+b1.area());

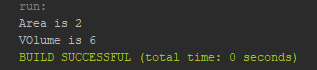
System.out.println("VOlume is "+b1.volume());

}

}

****

**OUTPUT:**

****

# lab Task 21: inheritance (BOX + WEIGht):

**CODE:**

package labtask21;

class box{

int height,width,depth;

box(){

width=-1;

height=-1;

depth=-1;

}

box(int h,int w,int d){

width=w;

height=h;

depth=d;

}

box(box obj){

width=obj.width;

height=obj.height;

depth=obj.depth;

}

int volume(){

return width\*height\*depth;}}

class boxweight extends box{

int weight;

boxweight(int h,int w,int d,int m){

super(h, w, d); weight=m;

}

boxweight(box obj,int m){

super(obj);weight=m;

}

}

public class LabTask21 {

public static void main(String[] args) {

boxweight b1=new boxweight(1,2,3,7);

boxweight b2=new boxweight(b1,4);

System.out.println("volume of box 1 is "+b1.volume());

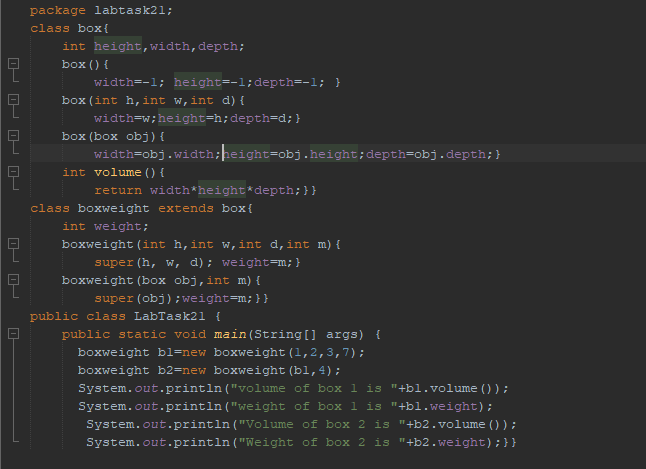
System.out.println("weight of box 1 is "+b1.weight);

System.out.println("Volume of box 2 is "+b2.volume());

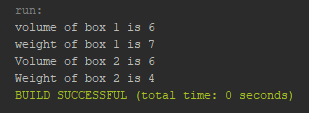
System.out.println("Weight of box 2 is "+b2.weight);

}

}

****

**OUTPUT:**

****

# Lab Task 22: Inheritance (Box model + Weight + Shipping):

**CODE:**

package labtask22;

class box{

double height,width,depth;

box(double h,double w,double d){ //when all dimensions are specified

height=h;width=h;depth=d;

}

box(box ob){ //for creating a clone of object

width=ob.width;

height=ob.height;

depth=ob.depth;

}

box(){ //for default or empty arguement

width=-1;

height=-1;

depth=-1;

}

box(double len){ //for cube as cube =l\*l\*l

depth=height=width=len;

}

double volume(){ //for returning volume

return width\*depth\*height;

}

}

class boxweight extends box{

double weight;

boxweight(double h,double w,double d,double m){ //when all dimensions are specified

super(h,w,d);weight=m;

}

boxweight(){ //for default

super();weight=-1;

}

boxweight(boxweight obj){ //for cloning

super(obj); weight=obj.weight;

}

boxweight(double len,double m){ //for cube

super(len);weight=m;

}

}

class boxweightship extends boxweight{

double cost;

boxweightship(double h,double w, double d, double m, double c){

super(h,w,d,m);cost=c;

}

boxweightship(){

super();cost=0;

}

boxweightship(boxweightship obj){

super(obj);cost=obj.cost;

}

boxweightship(double len,double m,double c){

super(len,m);cost=c;

}

}

public class LabTask22 {

public static void main(String[] args) {

boxweightship b1=new boxweightship();

boxweightship b2=new boxweightship(1,2,3,4,5);

boxweightship b3=new boxweightship(4,2,5);

boxweightship b4=new boxweightship(b3);

//FOR BOx OF DEFAULt ARGUEMENT

System.out.println("The volume of box 1 is "+b1.volume());

System.out.println("The weight of box 1 is "+b1.weight);

System.out.println("The shipping cost is "+b1.cost);

System.out.println();

//FOR BOX of ALL DIMENSIONS

System.out.println("The volume of box 2 is "+b2.volume());

System.out.println("The weight of box 2 is "+b2.weight);

System.out.println("The shipping cost is "+b2.cost);

System.out.println();

//FOR BOX OF CUBE

System.out.println("The volume of box 3 is "+b3.volume());

System.out.println("The weight of box 3 is "+b3.weight);

System.out.println("The shipping cost is "+b3.cost);

System.out.println();

//FOR BOX OF CLONE(BOX 3)

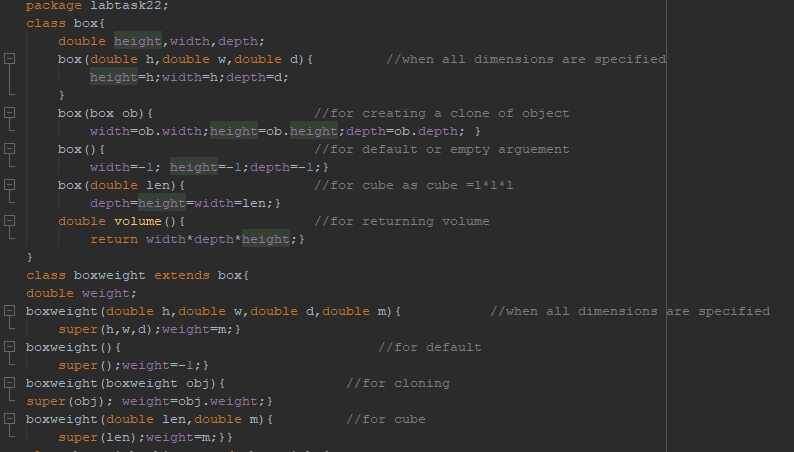
System.out.println("The volume of box 3 is "+b3.volume());

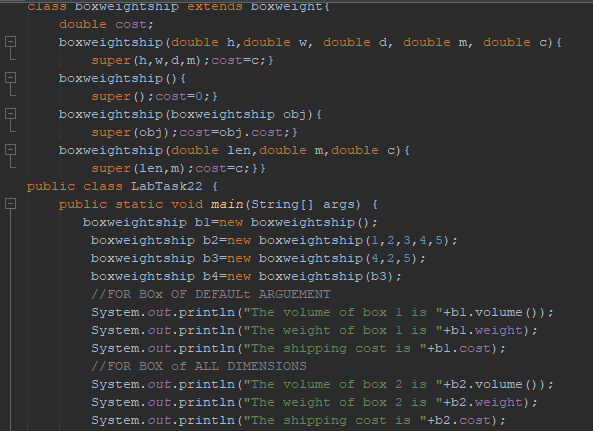
System.out.println("The weight of box 3 is "+b3.weight);

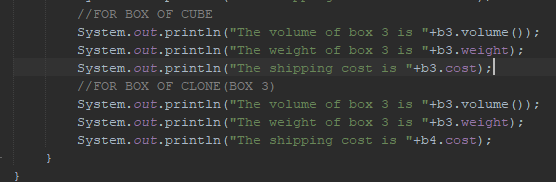
System.out.println("The shipping cost is "+b4.cost);

}

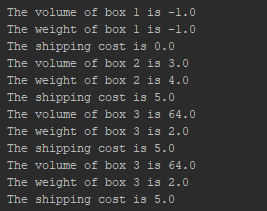
}







**OUTPUT:**

****

# Lab Task 23: use of THIS keyword

**CODE:**

package labtask23;

class A {

int a;

A(int a) {

this.a = a;

}

void show() {

System.out.println(a);

}

}

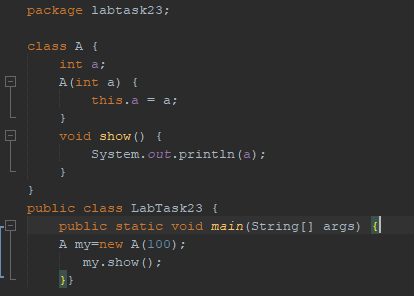
public class LabTask23 {

public static void main(String[] args) {

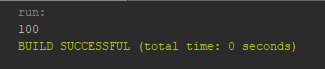
A my=new A(100);

my.show();

}}



**OUTPUT:**

****

# lab Task 24 : Static Example

**CODE:**

package labtask24;

class test{

static int staticcounter=-1;

int counter=-1;}

public class LabTask24 {

public static void main(String[] args) {

test t1=new test();

test t2=new test();

t2.counter =5;

t2.staticcounter = 5;

System.out.println("Value of test 1 counter : "+t1.counter);

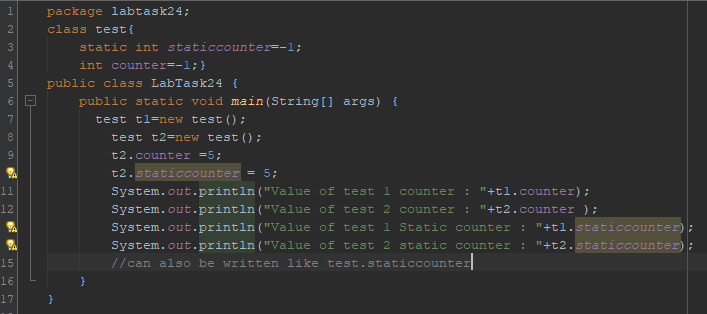
System.out.println("Value of test 2 counter : "+t2.counter );

System.out.println("Value of test 1 Static counter : "+t1.staticcounter);

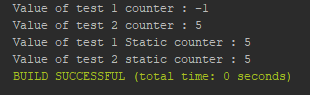
System.out.println("Value of test 2 static counter : "+t2.staticcounter); //can also be written like test.staticcounter

}

}



**OUTPUT:**

****

# lab Task 25 : Static Example (variables + methods):

**CODE:**

package labtask25;

class staticdemo{

static int a=90;

static int b=67;

static void callme(){

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

}

}

public class LabTask25 {

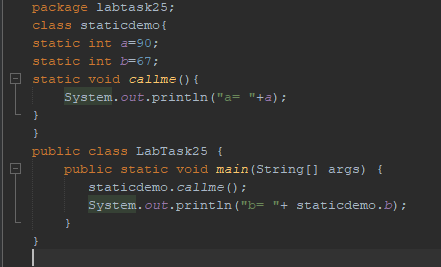
public static void main(String[] args) {

staticdemo.callme();

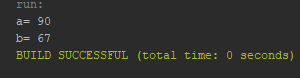
System.out.println("b= "+ staticdemo.b);

}

}



**OUTPUT:**

****

# lab Task 26 : ENCAPSULATED STUDENTS

**CODE:**

package labtask26;

class student{

private String name;

private int [] result=new int[5];

//CONSTRUCTOR

student(String n, int [] arr){

result=arr;

name=n;}

double average(){

double avg=(result[0]+result[1]+result[2]+result[3]+result[4])/5;

return avg; }

int [] result(){

return result; }

String name(){

return name;}

}

public class LabTask26 {

public static void main(String[] args) {

student st1=new student("Farooq Ali",new int[]{20,40,60,40,30});

student st2=new student("Ali",new int[]{30,45,67,5,4});

//Student 1 Marks

System.out.println(st1.name()+" 's Average Marks are : "+st1.average());

//STUDENT 2 MARKS

System.out.println(st2.name()+" 's Average Marks are : "+st2.average());

//STUDENT 1 And 2 COMMPARISION

if(st1.average()>st2.average()){

System.out.println(st1.name()+"'s Marks are higher");

}

else{

System.out.println(st2.name()+"'s Marks are higher");

}

//STUDENT 3 MARKS having name of first student and result of 2nd student

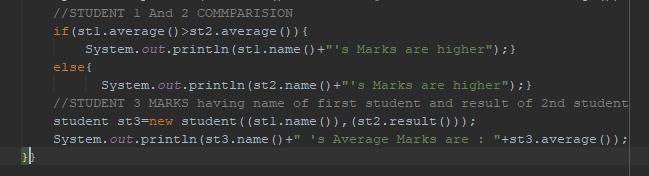
student st3=new student((st1.name()),(st2.result()));

System.out.println(st3.name()+" 's Average Marks are : "+st3.average());

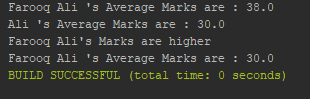
}

}





**OUTPUT:**

****

# lab Task 27 : Encapsulation

**CODE:**

package labtask27;

class encapsulation{

private int m1;

private int m2;

private int m3;

public void setm1(int newvalue){

m1=newvalue;}

public void setm2(int newvalue){

m2=newvalue;}

public void setm3(int newvalue){

m3=newvalue;}

public int getm1(){

return m1; }

public int getm2(){

return m2;}

public int getm3(){

return m3;}}

public class LabTask27 {

public static void main(String[] args) {

encapsulation obj=new encapsulation();

obj.setm1(34);

obj.setm2(45);

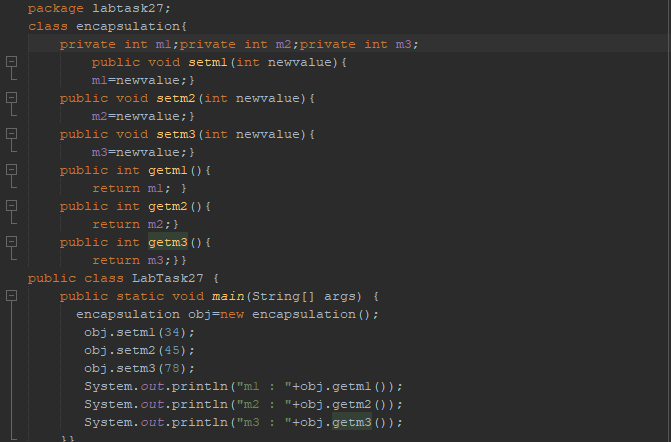
obj.setm3(78);

System.out.println("m1 : "+obj.getm1());

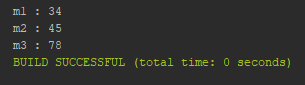
System.out.println("m2 : "+obj.getm2());

System.out.println("m3 : "+obj.getm3());

}}



**OUTPUT:**

****

# lab Task 28 : Abstract

**CODE:**

package labtask28;

abstract class animal{

public abstract void sound();

}

class cat extends animal{

@Override

public void sound(){

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

}

}

class dog extends animal{

@Override

public void sound(){

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

}

}

public class LabTask28 {

public static void main(String[] args) {

dog d1=new dog();

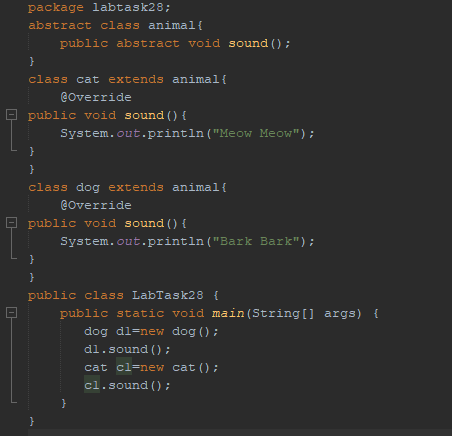
d1.sound();

cat c1=new cat();

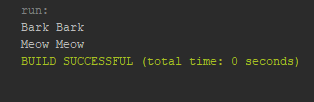
c1.sound();

}

}



**OUTPUT:**

****

# lab Task 29: Method Overloading

**CODE:**package labtask.pkg29;

class demoOverload {

void test() {

System.out.println("no parameter ");

}

void test(int z) {

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

}

void test(int x, int y) {

System.out.println("value of x,y is " + x + " " + y);

}

double test(double z) {

System.out.println("double z is " + z);

return z \* z;

}

}

public class LabTask29 {

public static void main(String[] args) {

demoOverload demo = new demoOverload();

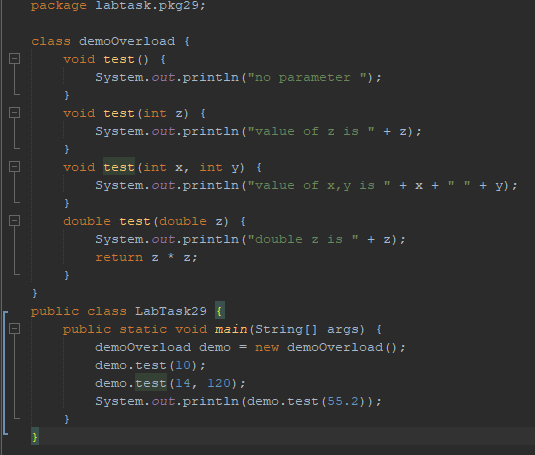
demo.test(10);

demo.test(14, 120);

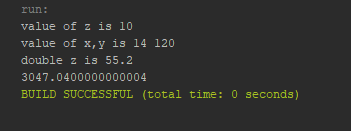
System.out.println(demo.test(55.2));

}

}



**OUTPUT:**

****

# lab Task 30 : Static and Instance Blocks

**CODE:**

package labtask30;

class test{

// Static Block

static{

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

}

// Non - Static / Instance Block

{

System.out.println("Non static / Instance block");

}

// Constructor

test(){

System.out.println("My class constructor");

}

}

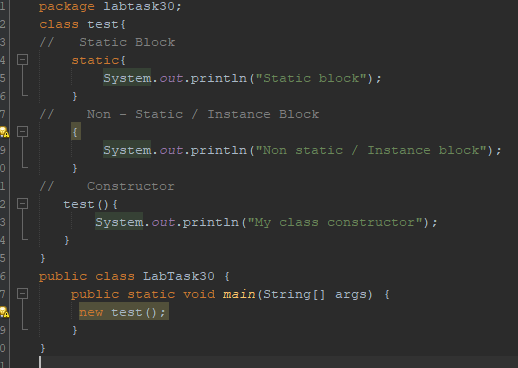
public class LabTask30 {

public static void main(String[] args) {

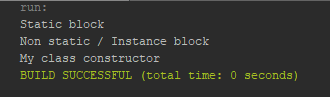
new test();

}

}



**OUTPUT:**

****

# lab Task 31: Final Keyword

**CODE:**

package labtask31;

class bike{

final int speedlimit;

bike(){

speedlimit=80;

System.out.println(speedlimit);

}

}

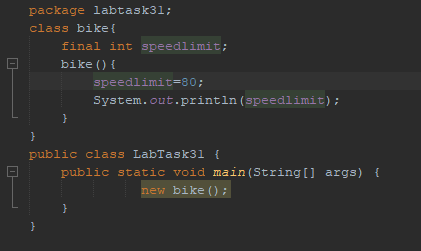
public class LabTask31 {

public static void main(String[] args) {

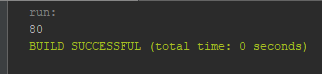
new bike();

}

}

****

**OUTPUT:**

****

# lab Task 32: Dynamic Method Dispatch

**CODE:**

package labtask32;

class A{

void show(){

System.out.println("in class A Method"); }}

class B extends A{

@Override

void show(){

System.out.println("In class b");}}

class C extends A{

@Override

void show(){

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

public class LabTask32 {

public static void main(String[] args) {

A r;

B b = new B();

C c = new C();

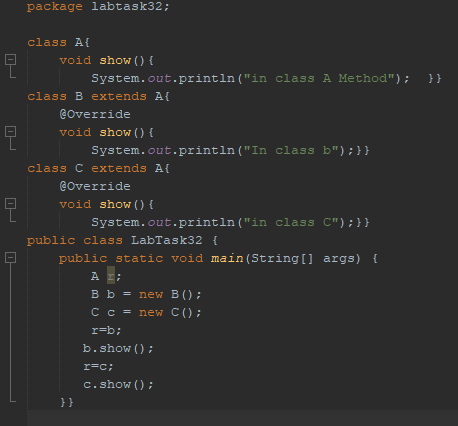
r=b;

b.show();

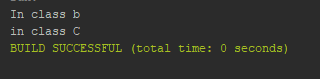
r=c;

c.show();

}}



**OUTPUT:**

****

# lab Task 33: Car Details Using Scanner

**CODE:**

package labtask33;

import java.util.Scanner;

class Car{

String name;

String model;

String owner;

int price\_in\_millions;

void assignValues(){

Scanner sc=new Scanner(System.in);

System.out.print("Enter Car Name :\t");

name = sc.nextLine();

System.out.print("Enter Car Model:\t");

model = sc.nextLine();

System.out.print("Enter Car Owner:\t");

owner = sc.nextLine();

System.out.print("Enter Car Price(in Millions):\t");

price\_in\_millions = sc.nextInt();

}

void display(){

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

System.out.println("Car Model :\t"+model);

System.out.println("Car Owner :\t"+owner);

System.out.println("Car price:\t"+price\_in\_millions+" Million PKR");

}

}

public class LabTask33 {

public static void main(String[] args) {

Car car1=new Car();

Car car2=new Car();

Car car3=new Car();

car1.assignValues();

car2.assignValues();

car3.assignValues();

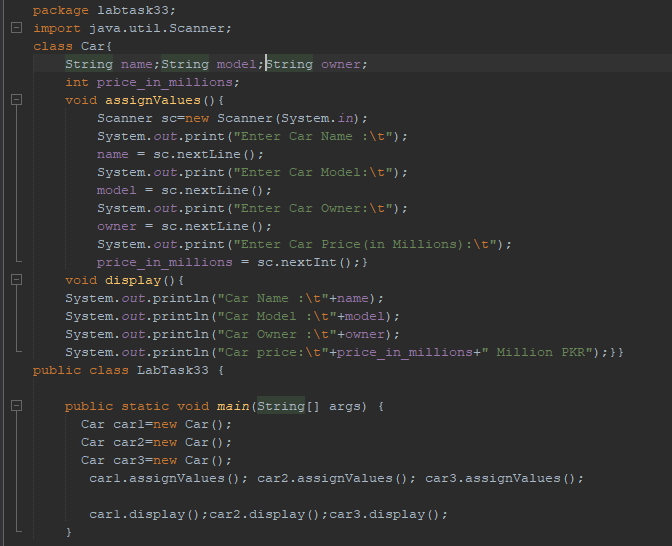
car1.display();

car2.display();

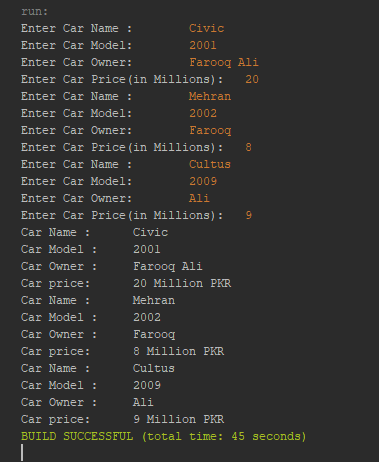
car3.display();

}

}



**OUTPUT:**

****

# lab Task 34: Car Example Using Constructor

**CODE:**

package labtask34;

class car{

String carName;

String carOwner;

int carModel;

float carPrice;

void setvalues(String n, String o, int m, int p){

carName=n;

carOwner=o;

carModel=m;

carPrice=p;

}

void display(){

System.out.println("Car Name: "+carName);

System.out.println("Car Owner Name: "+carOwner);

System.out.println("Car Model: "+carModel);

System.out.println("Car Price "+carPrice);}}

public class LabTask34 {

public static void main(String[] args) {

car car1=new car();

car car2=new car();

car car3=new car();

car1.setvalues("Honda", "Farooq", 1998, 200000);

car1.display();

car2.setvalues("Suzuki", "Farooq", 2004, 100000);

car2.display();

car3.setvalues("toyota", "Farooq", 2009 , 400000);

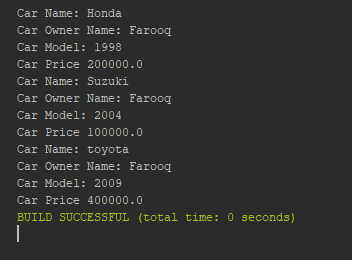
car3.display();

}

}



**OUTPUT:**

****

# Lab Task 35: CIRCLE EXAMPLE

**CODE:**

package labtask35;

class circle{

double radius;

double pi;

circle(){

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

}

circle(double a , double b){

radius=a;

pi=b;

}

void circum(){

System.out.println("Cirumeference : "+(2\*pi\*radius));

}

}

public class LabTask35 {

public static void main(String[] args) {

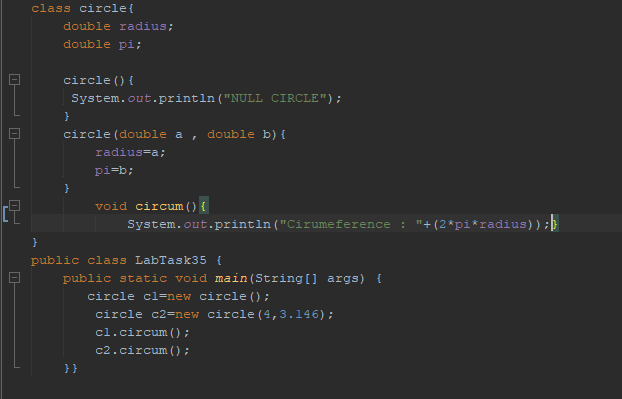
circle c1=new circle();

circle c2=new circle(4,3.146);

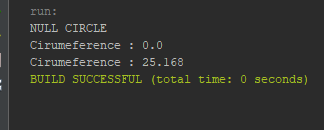
c1.circum();

c2.circum();

}}



**OUTPUT:**

****

# Lab task 36: Savings Account

**CODE:**

package labtask36;

class SavingsAccount{

static double annualInterestRate;

private double savingsBalance;

SavingsAccount(double s){

savingsBalance = s;

}

void calculateMonthlyInterest(){

double monthlyInterest = (savingsBalance \* annualInterestRate/100)/12;

savingsBalance+= monthlyInterest;

System.out.println("Savings Balance "+monthlyInterest);

}

double getNewSB(){

return savingsBalance;

}

static void modifyInterestRate(double a){

annualInterestRate = a;}}

public class LabTask36 {

public static void main(String[] args) {

SavingsAccount saver1 = new SavingsAccount(2000);

SavingsAccount.annualInterestRate= 3;

saver1.calculateMonthlyInterest();

System.out.println("New savings balance is "+saver1.getNewSB());

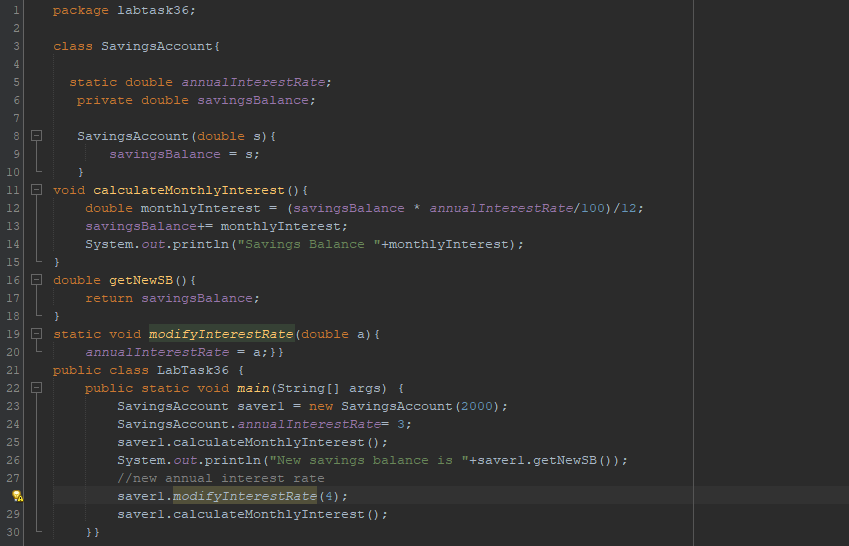
//new annual interest rate

saver1.modifyInterestRate(4);

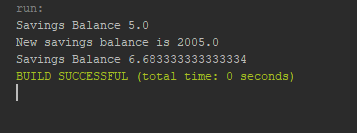
saver1.calculateMonthlyInterest();

}

}



**OUTPUT:**

****

# Lab Task 37: TIME TASK

**CODE:**

package labtask37;

class time {

int hrs;int mins;int secs;

boolean check;

time(int h, int m, int s) {

hrs = h;mins = m;secs = s;

}

void checktime() {

if (hrs <= 24 && hrs >= 0) {

if (mins <= 60 && mins >= 0) {

if (secs <= 60 && secs >= 0) {

check = true;

} else {

System.out.println("Invalid Second Input");

check = false;}

} else {

System.out.println("Invalid Minutes Input");

check = false; }

} else {

System.out.println("Invalid Hours Input");

check = false;

}}

void display() {

if (check == true) {

System.out.println(hrs + " hours " + mins + " minutes " + secs + " seconds");

}}}

public class LabTask37 {

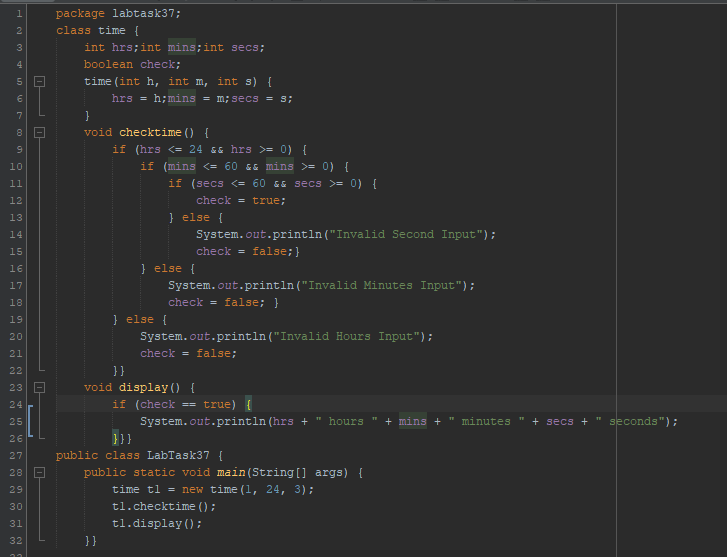
public static void main(String[] args) {

time t1 = new time(1, 24, 3);

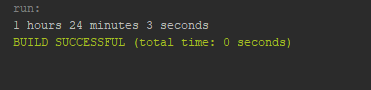
t1.checktime();

t1.display();

}}



**OUTPUT:**

****

# LAB TASK 38: CLOCK Task

**CODE:**

package labtask38;

import java.time.LocalTime;

import java.time.format.DateTimeFormatter;

abstract class clock{

int hr;

int min;

int sec;

boolean check=true;

void display(){

DateTimeFormatter Formatter = DateTimeFormatter.ofPattern("HH:mm:ss");

LocalTime time =LocalTime.now();

hr = time.getHour();

min = time.getMinute();

sec = time.getSecond(); }

}

class overridenClock extends clock{

@Override

void display(){

super.display();

System.out.println("24-H : "+hr+" : "+min+" : "+sec);

String ap = "AM";

if (hr>12){

ap="PM";

hr-=12;}

System.out.println("Standard Format : "+hr+" : "+min+" : "+sec+" "+ap);}}

public class LabTask38 {

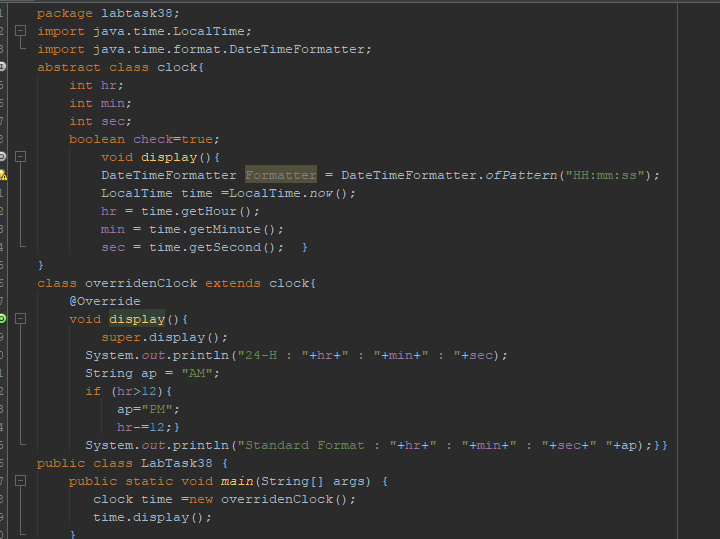
public static void main(String[] args) {

clock time =new overridenClock();

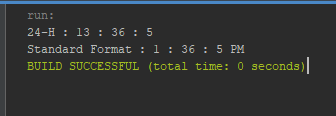
time.display();

}

}



**OUTPUT:**

****

# Lab task 39: Abstract class and Method

**CODE:**

package labtask39;

abstract class A{

abstract void callme();

{

System.out.println("This is a concrete Method");

}

}

class B extends A{

@Override

void callme(){

System.out.println("B's Implementation of Call Me");

}

}

public class LabTask39 {

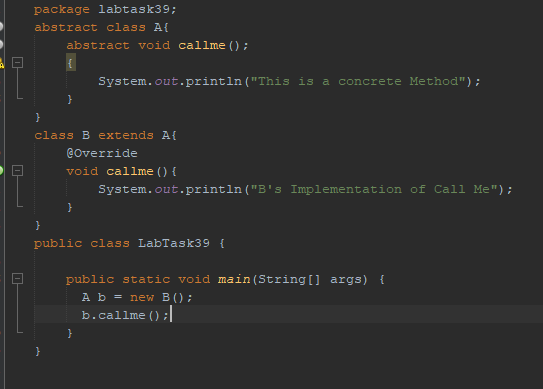
public static void main(String[] args) {

A b = new B();

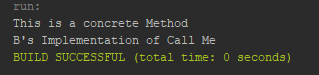
b.callme();

}

}



**OUTPUT:**

****

# Lab Task 40: Employee Task

**CODE:**

package labtask40;

import java.util.Scanner;

import java.time.LocalDate;

import java.time.format.DateTimeFormatter;

import java.time.temporal.ChronoUnit;

class employee{

String employName;

int employCode;

String employDOJ;

String currentdate;

void getvalues(){

Scanner s=new Scanner(System.in);

System.out.print("Enter Employ Name :\t");

employName=s.nextLine();

System.out.print("Enter Employee Code :\t");

employCode=s.nextInt();

s.nextLine();

System.out.print("Enter Employee Date Of Joining (dd/MM/yyyy) : \t");

employDOJ=s.nextLine();

System.out.print("Enter Current Date :\t");

currentdate=s.nextLine();

}

void checkTenure(){

DateTimeFormatter formatter=DateTimeFormatter.ofPattern("dd/MM/uuuu");

LocalDate doj=LocalDate.parse(employDOJ, formatter);

LocalDate curdate=LocalDate.parse(currentdate, formatter);

long yearsBetween=ChronoUnit.YEARS.between(doj, curdate);

if(yearsBetween>=3){

System.out.println(employName+"'s Tenure is greater than 3 years");

}

else{

System.out.println(employName+"'s Tenure is less than 3 years");

}

}

}

public class LabTask40 {

public static void main(String[] args) {

employee emp1=new employee();

employee emp2=new employee();

System.out.println("ENTER EMPLOYEE 1 DETAILS");

emp1.getvalues();

System.out.println("ENTER EMPLOY 2 DETAILS");

emp2.getvalues();

System.out.println("EMPLOY 2 DETAILS:\t");

System.out.println("Employ Name :\t"+emp1.employName);

System.out.println("Employ Code :\t"+emp1.employCode);

System.out.println("Employ Date of Joining :\t"+emp1.employDOJ);

System.out.println("EMPLOY 2 DETAILS:\t");

System.out.println("Employ Name :\t"+emp2.employName);

System.out.println("Employ Code :\t"+emp2.employCode);

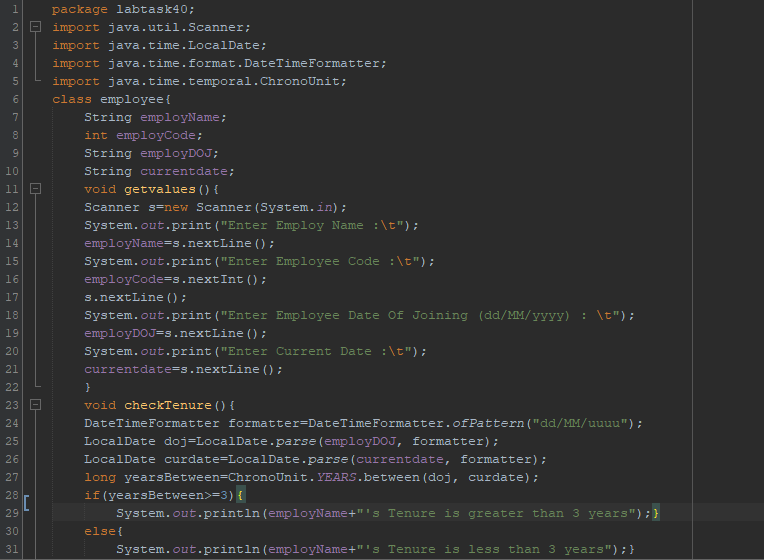
System.out.println("Employ Date of Joining :\t"+emp2.employDOJ);

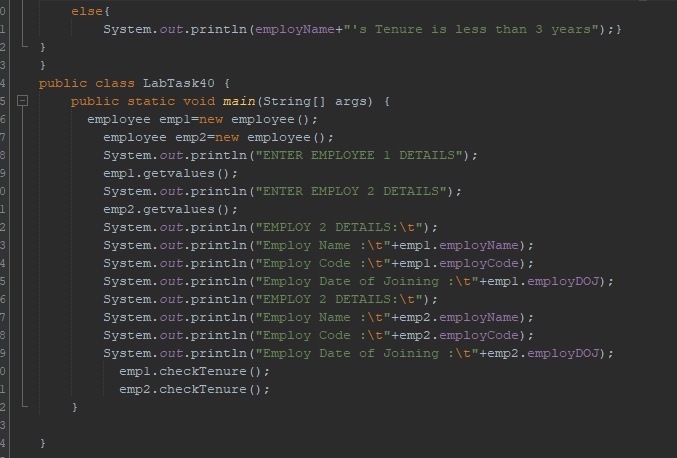
emp1.checkTenure();

emp2.checkTenure();

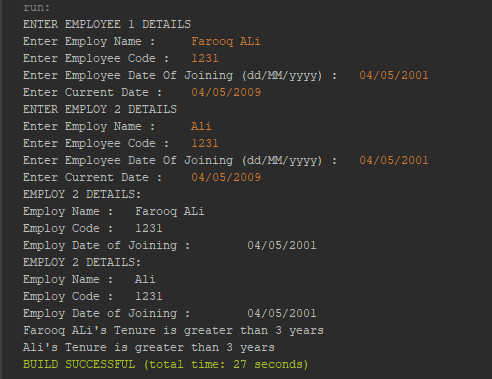
}

}





**OUTPUT:**

****

# Lab Task 41: Task Game

**CODE:**

package labtask41;

class game{

public void type(){

System.out.println("Indoor & Outdoor");

}

}

class cricket extends game{

public void type(){

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

}

}

public class LabTask41 {

public static void main(String[] args) {

game gm = new game();

cricket ck = new cricket();

gm.type();

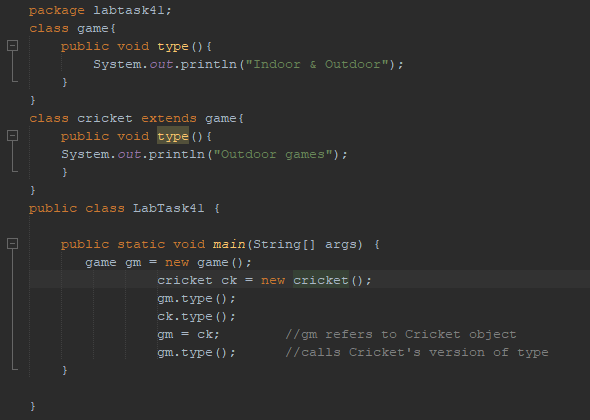
ck.type();

gm = ck; //gm refers to Cricket object

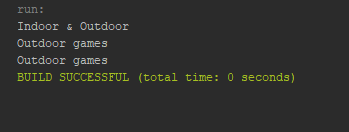
gm.type(); //calls Cricket's version of type

}

}



**OUTPUT:**

****

# Lab Task 42: Interface Month Numbers

**CODE:**

package labtask42;

interface MonthNumbers {

public static final int JANUARY = 1,FEBRUARY = 2,MARCH = 3, APRIL = 4,

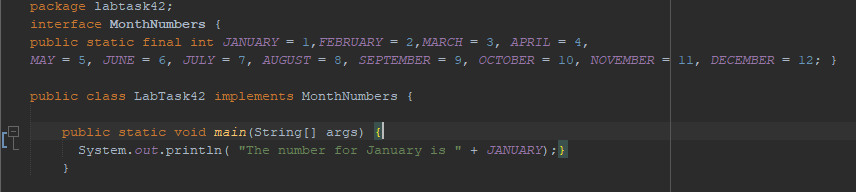
MAY = 5, JUNE = 6, JULY = 7, AUGUST = 8, SEPTEMBER = 9, OCTOBER = 10, NOVEMBER = 11, DECEMBER = 12; }

public class LabTask42 implements MonthNumbers {

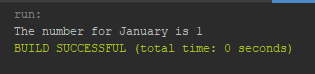
public static void main(String[] args) {

System.out.println( "The number for January is " + JANUARY);}

}



**OUTPUT:**

****

# Lab Task 43: Interface Register For Exams

**CODE:**

package labtask43;

interface RegisterForExams{

public void register();

}

class student implements RegisterForExams{

String name; int rollNo;double Gpa;

student(String n, int r, double g){

name =n;rollNo=r;Gpa =g;}

@Override

public void register(){

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

System.out.println("Student Roll NO : "+rollNo );

System.out.println("Student GPA : "+Gpa); }}

class employee implements RegisterForExams{

String name; int code;

employee(String n, int c){

name = n;code = c;}

@Override

public void register(){

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

System.out.println("Employee Code : "+code );}}

class interfaceTest{

void testApp(RegisterForExams a){

a.register();}}

public class Labtask43 {

public static void main(String[] args) {

interfaceTest t1 = new interfaceTest();

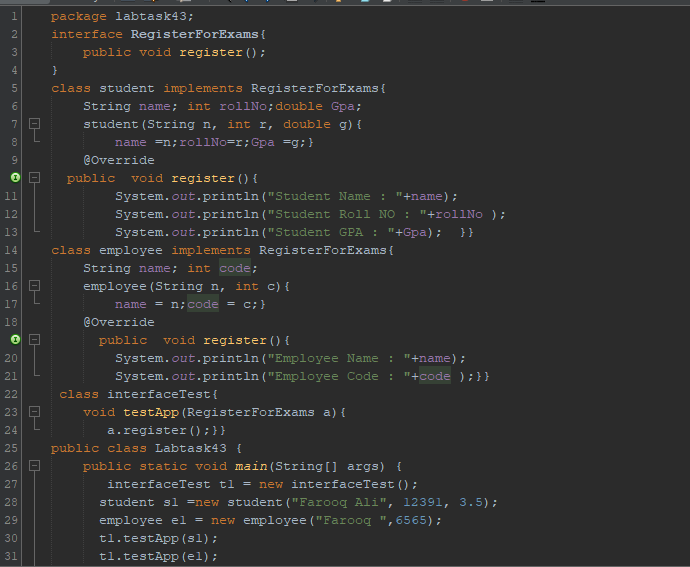
student s1 =new student("Farooq Ali", 12391, 3.5);

employee e1 = new employee("Farooq ",6565);

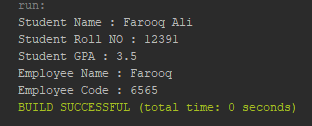
t1.testApp(s1);

t1.testApp(e1);

}}



**OUTPUT:**

****

**THE END**