**Assignment 2**

**1. Define a class named Course having data members ID, Description, Duration and**

**Fees. The class should have one parametrized constructors and GetData() function**

**member to display the data.**

**-Create an array of 5 course objects and then display the data for all of them.**

class Course{

int id;

String descript;

int duration;

int fees;

Course(int id,String descript,int duration,int fees){

this.id=id;

this.descript=descript;

this.duration=duration;

this.fees=fees;

}

void GetData(){

System.out.println("Course id:"+id+"\nCourse Description:"+descript+"\nCourse Durartion:"+duration+"\nCourse Fees:"+fees);}

public static void main(String[]args){

Course co=new Course(1001,"B.Tech",6,78900);

co.GetData();

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

Course c[] = new Course[5];

c[0]=new Course(1002,"DITISS",4,8000);

c[1]=new Course(1003,"DAC",4,790870);

c[2]=new Course(1004,"DBDA",4,78900);

c[3]=new Course(1005,"DESD",4,90000);

c[4]=new Course(1006,"DMC",4,2800000);

System.out.println("Course Object 1 data:");

//System.out.println();

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

c[0].GetData();

System.out.println();

System.out.println("Course Object 2 data:");

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

c[1].GetData();

System.out.println();

System.out.println("Course Object 3 data:");

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

c[2].GetData();

System.out.println();

System.out.println("Course Object 4 data:");

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

c[3].GetData();

System.out.println();

System.out.println("Course Object 5 data:");

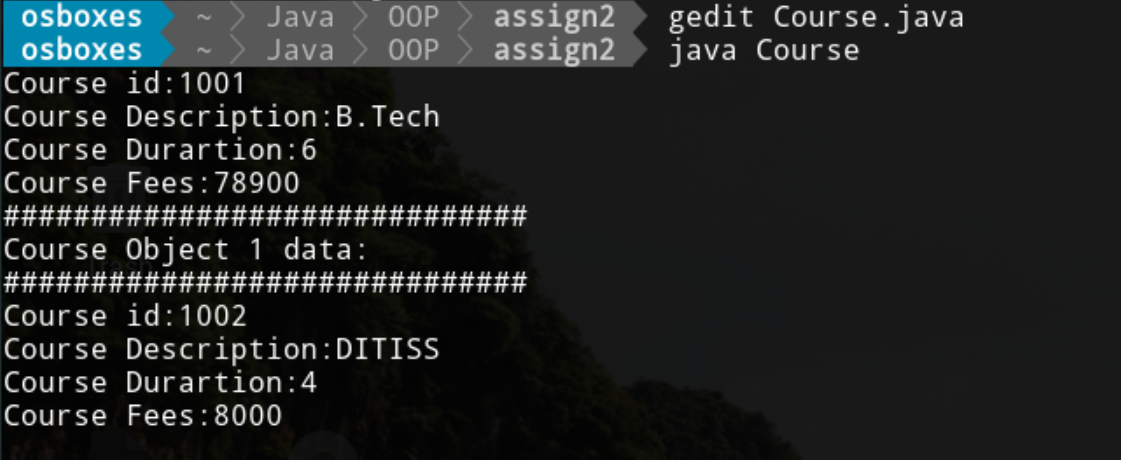
System.out.println("##############################");

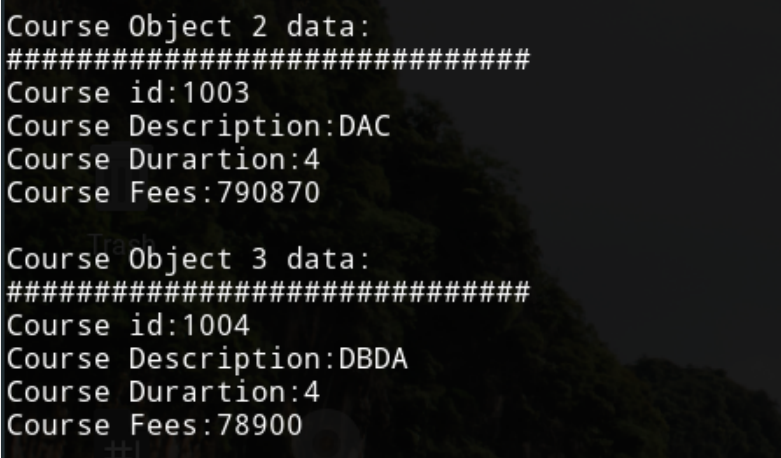
c[4].GetData();

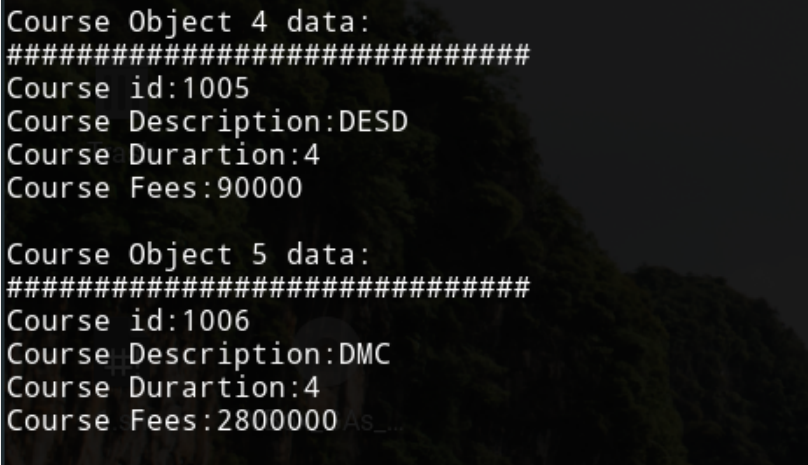
System.out.println();

}

}







**2. Modify program 1 to add a default constructor and a SetData() member**

**function.**

**-Create an array of 3 student using the default constructor and another array of 2 students using the parametrized constructor, and then display the data of all 5 course objects.**

class Student{

int id;

String descript;

int duration;

int fees;

//default constructor

Student(){}

//setData() method

void setData(int id,String descript,int duration,int fees){

this.id=id;

this.descript=descript;

this.duration=duration;

this.fees=fees;

}

Student(int id,String descript,int duration,int fees){

this.id=id;

this.descript=descript;

this.duration=duration;

this.fees=fees;

}

void getData(){

System.out.println("Course id:"+id+"\nCourse Description:"+descript+"\nCourse Durartion:"+duration+"\nCourse Fees:"+fees);}

public static void main(String[]args){

Student s=new Student(100,"DESD",6,900000);

Student arr[]=new Student[3];

System.out.println("Course Object 1 Data:");

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

s.setData(101,"DAC",6,9000);

s.getData();

System.out.println();

System.out.println("Course Object 2 Data:");

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

s.setData(102,"DMC",6,100000);

s.getData();

System.out.println();

System.out.println("Course Object 3 Data:");

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

s.setData(103,"DBDA",6,15000);

s.getData();

System.out.println();

Student [] arr1=new Student[2];

System.out.println("Course Object 4 Data:");

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

arr1[0]=new Student(104,"DITISS",6,12000);

arr1[0].getData();

System.out.println();

System.out.println("Course Object 5 Data:");

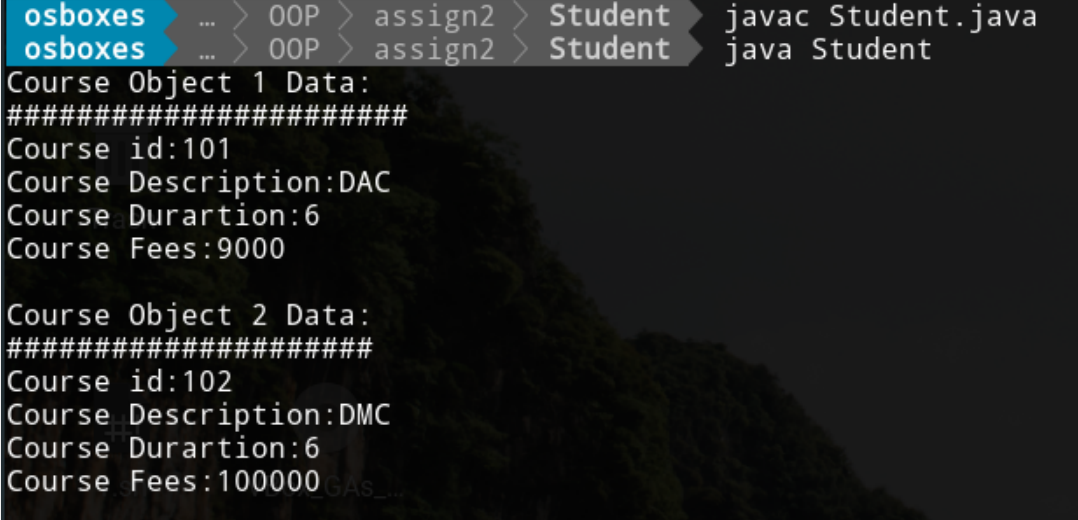
System.out.println("######################");

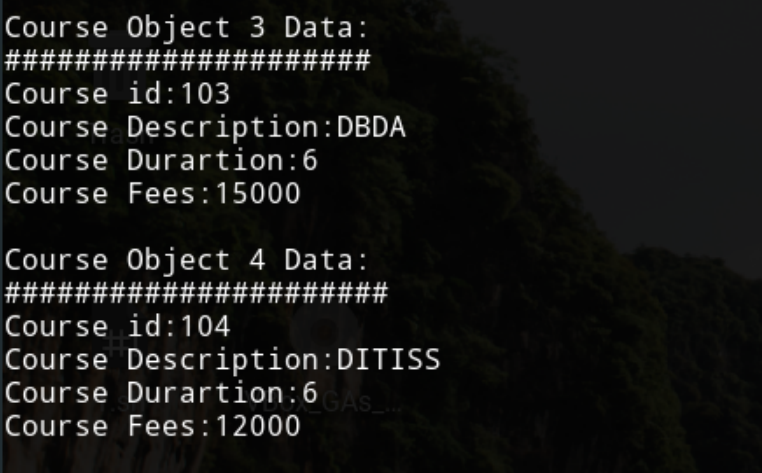
arr1[1]=new Student(105,"DASSD",7,15000);

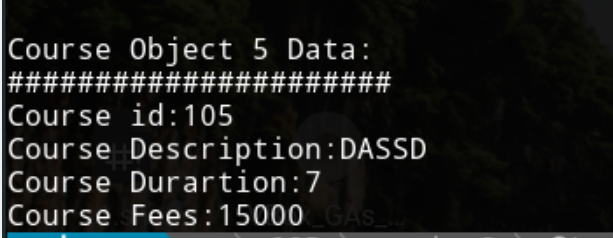
arr1[1].getData();

}

}

****

****

****

**3. Write a program to define constructors and finalize of a class and construct 3 objects and then show that it is not a guarantee that finalizer will be invoked for each object.**

class Finalize{

String name;

int id;

//parameterized constructor

Finalize(String name, int id){

this.name=name;

this.id=id;

}

//finalize method

public void finalize()

{

System.out.println("Garbage collector invoked!!!!");

}

void display() {

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

}

public static void main(String[] args) {

Finalize f1=new Finalize("Prachi",10);

Finalize f2=new Finalize("Swati",11);

Finalize f3=new Finalize("Nonu",12);

f1.display();

f2.display();

f3.display();

f1=null;

f2=null;

f3=null;

System.gc();

System.out.println("Objects are unreferenced");

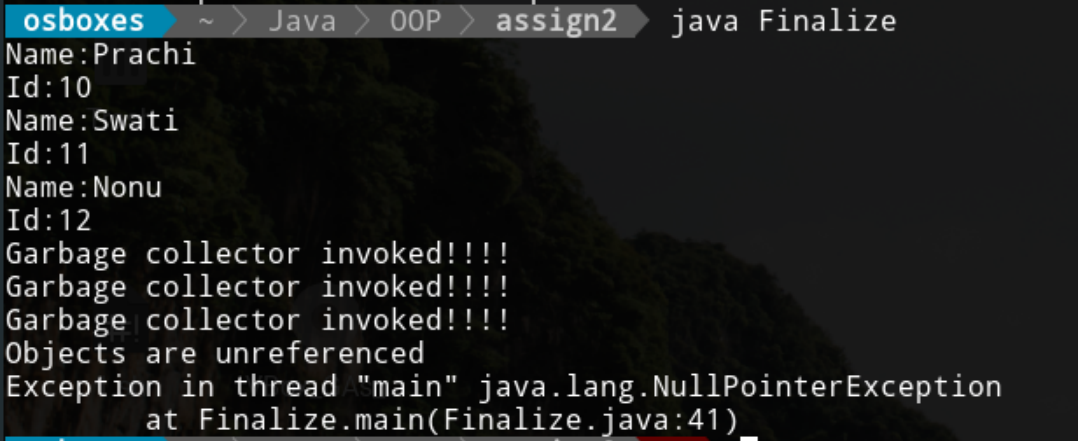
f1.display(); //it is only for check reference variable is exists or not

f2.display();

f3.display();

}

}



**4 Write a program to demonstrate the use of final keyword with**

**a) class**

final class A{ //final class

int a=10;

}

class Final extends A{

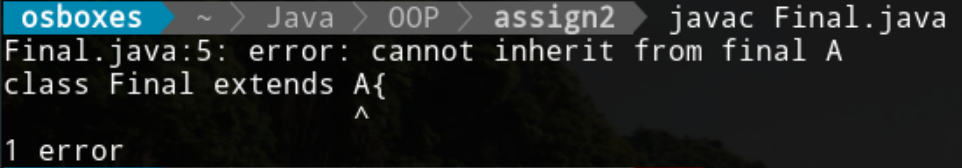
public static void main(String []args){

Final f=new Final();

System.out.println(f.a);}

}

}



**b) method**

class A{

int a=10;

final void A(){

System.out.println("Hello!!! I am in class A");

}

}

class Final extends A{

//ovveride the parent class method

void A(){

System.out.println("hello!! I am final");}

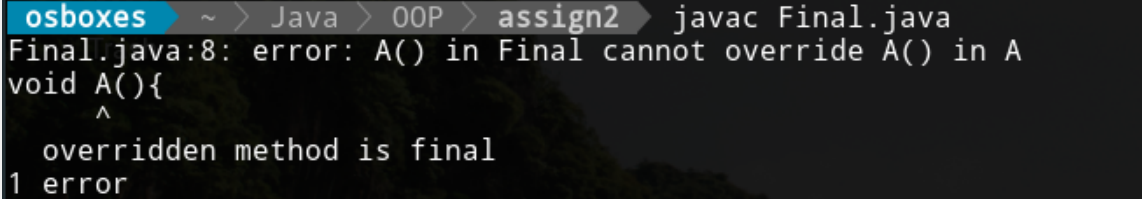
public static void main(String []args){

Final f=new Final();

f.A(); //call the method

System.out.println(f.a);}

}



**c) data member(primitive value and reference variable and show that you**

**can not refer this reference variable to other objects but can change the data**

**field of a final reference variable)**

public class FinalVar{

final int x = 10;

public static void main(String[] args) {

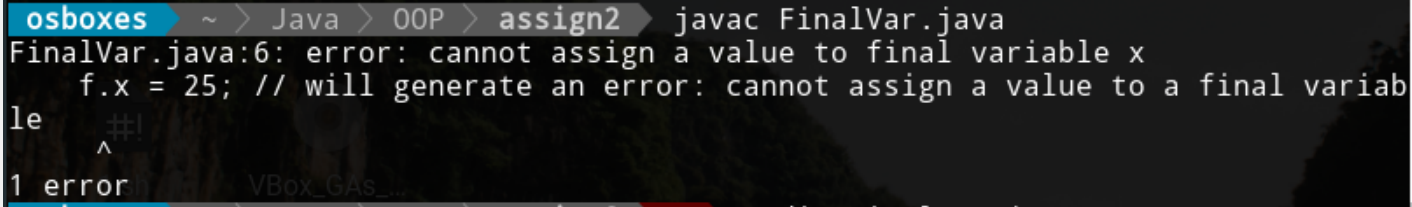
FinalVar f = new FinalVar();

f.x = 25;

System.out.println(f.x);

}

}



public class Student

{

int rollNo;

String name;

Student(int r,String n){ //constructor

this.rollNo=r;

this.name=n;

}

void display(){ //method to display data

System.out.println("Student RollNo:"+rollNo+"\nStudent Name:"+name);

}

public static void main(String[]args){

final Student s = new Student(102,"Prachi");

s.display();

Student s1 = new Student(103,"Vish");

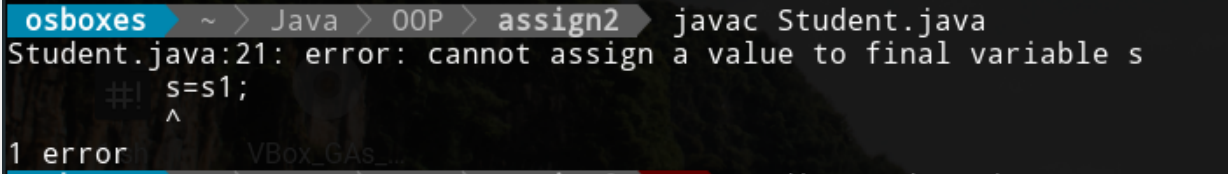
s1.display();

// object is a reference variable with final keyword so we can't assign it

s=s1;

}

}



public class Student

{

int rollNo;

String name;

public void setData(int rollNo,String name) {

this.rollNo=rollNo;

this.name = name;

}

void display(){

System.out.println("Student RollNo:"+rollNo+"\nStudent Name:"+name);

}

public static void main(String[]args){

final Student s = new Student();

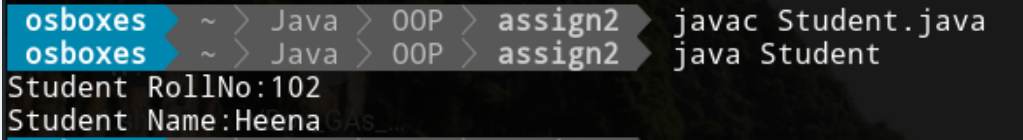
s.setData(101,"Prachi");

s.setData(102,"Heena");

s.display();

}

}



Yes we can change value of final reference object variable

**5. write a program to demonstrate the use of following operators.**

**a)right shift with sign bit operator >>**

import java.util.Scanner;

public class Operator {

public static void main(String[] args)

{ Scanner s=new Scanner(System.in);

System.out.print("Enter the value of a:");

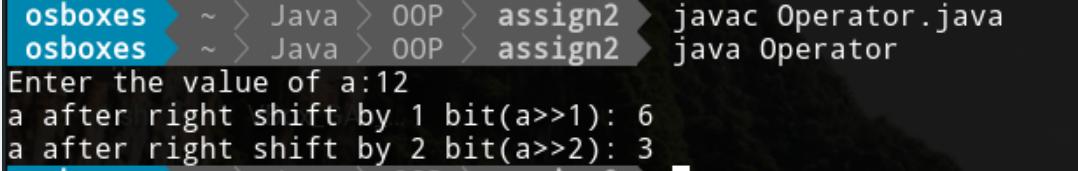
int a = s.nextInt();

System.out.println("a after right shift by 1 bit(a>>1): " + (a >> 1));

System.out.println("a after right shift by 2 bit(a>>2): " + (a >> 2));

}

}



**b)left shift operator <<**

import java.util.Scanner;

public class Operator {

public static void main(String[] args)

{ Scanner s=new Scanner(System.in);

System.out.print("Enter the value of a:");

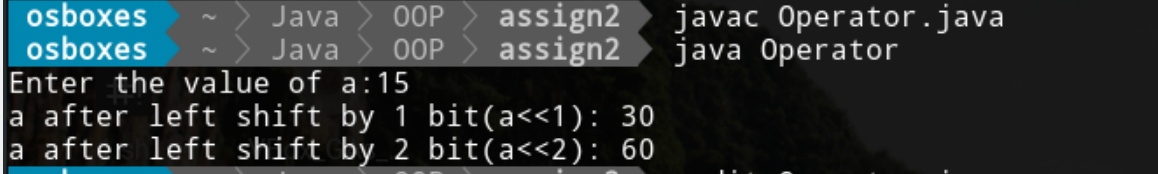
int a = s.nextInt();

System.out.println("a after left shift by 1 bit(a<<1): " + (a<<1));

System.out.println("a after left shift by 2 bit(a<<2): " + (a<<2));

}

}



**c)right shift with zero fill operator >>>**

import java.util.Scanner;

public class Operator {

public static void main(String[] args)

{ Scanner s=new Scanner(System.in);

System.out.print("Enter the value of a:");

int a = s.nextInt();

//System.out.println("a after left shift by 1 bit(a<<1): " + (a<<1));

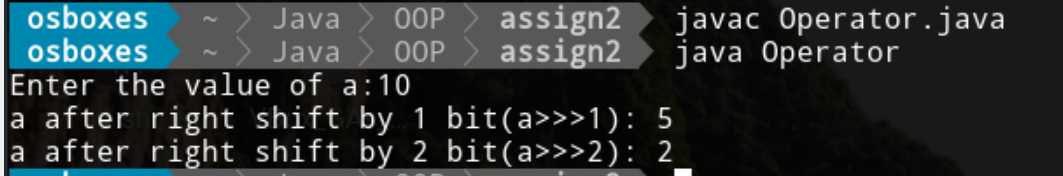
//System.out.println("a after left shift by 2 bit(a<<2): " + (a<<2));

System.out.println("a after right shift by 1 bit(a>>>1): " + (a>>>1));

System.out.println("a after right shift by 2 bit(a>>>2): " + (a>>>2));

}

}



**6. write a program to demonstrate**

**a) Labeled break**

class Break

{

public static void main(String [] args)

{

int i=4;

label:

while(i<20)

{

if(i==10)

break label;

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

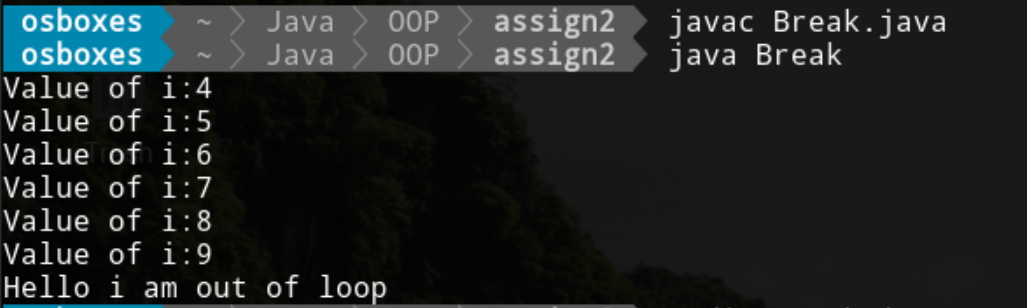
i++;

}

System.out.println("Hello i am out of loop");

}

}



**b) Labeled continue**

class Break

{

public static void main(String [] args)

{

int i=4;

label:

while(i<20)

{

if(i==10)

continue label;

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

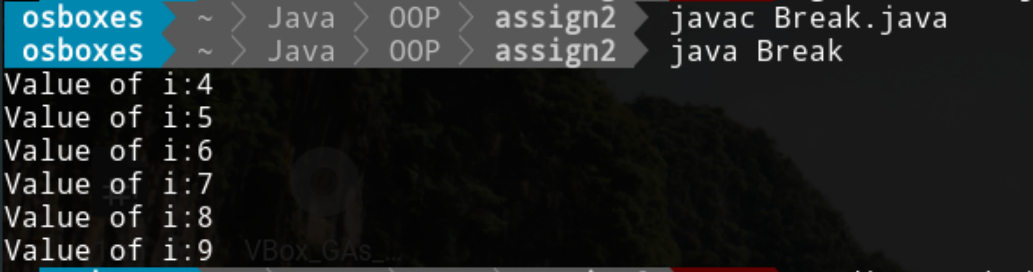
i++;

}

System.out.println("Hello I am out of loop");

}

}



**7. Demonstrate the use of ‘this’ keyword**

**a) To refer to current object.**

class This

{

int a;

int b;

// Parameterized constructor

This(int a, int b)

{

this.a = a; //refer current object

this.b = b;

}

void display()

{

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

}

public static void main(String[] args)

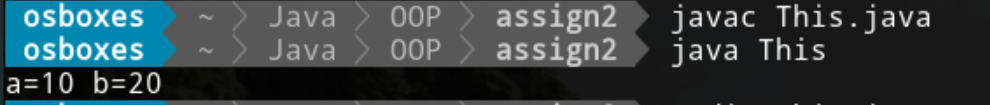
{

This T = new This(10,20);

T.display();

}

}



**b) Inside a constructor to call another constructor.**

class This

{

int a;

int b;

This(){

System.out.println("Hello!! I am a default Constructor");

}

// Parameterized constructor

This(int a, int b)

{ this(); //call constructor

this.a = a; //refer current object

this.b = b;

}

void display()

{

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

}

public static void main(String[] args)

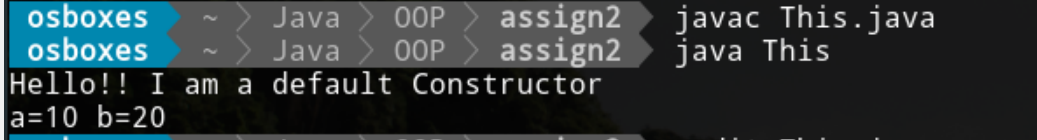
{

This T = new This(10,20);

T.display();

}

}



**And also show that this can not be used in static context area.**

class Static{

static int a=2000;

static int b=1000;

Static(int a,int b){

this.a=a;

this.b=b;}

static void data(){

System.out.println("hello i am static method");

}

void display(){

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

}

public static void main(String[]args){

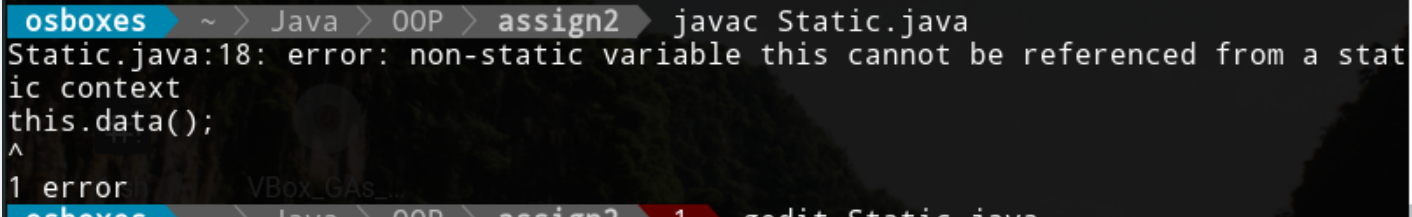
Static s=new Static(10,1000);

s.display();

this.data(); //call static method using this

}

}



**8. Demonstrate the use of ‘super’ keyword.**

**a) To refer to a member of super class.**

class Employee{

int id=1000;

static String org="C-DAC";

}

class Teacher extends Employee{

int id=5000;

void display(){

System.out.println("Id of Employee:"+super.id+"\nId of Teacher:"+id);

System.out.println("Teacher Organisation:"+org);}

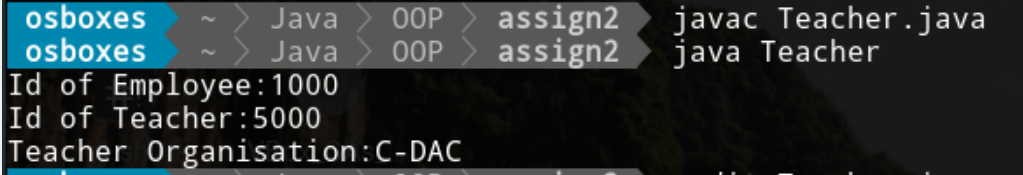
public static void main(String[]args){

Teacher t=new Teacher();

t.display();

}

}



**b) To call super class constructor from sub class constructor.**

class Employee{

int id;

static String org="C-DAC";

//constructor of parent class

Employee(int id){

System.out.print("Employee Id:");

System.out.println(this.id=id);

System.out.println("Employee Org:"+org);

}

}

class Teacher extends Employee{

int id;

int age;

//constructor of child class

Teacher(int id,int age){

super(1000); //super() use to call parent class constructor in child class

this.id=id;

this.age=age;

}

void display() //to display data

{

System.out.println("Teacher Id:"+id+"\nTeacher Age:"+age);

System.out.println("Teacher Org:"+org);}

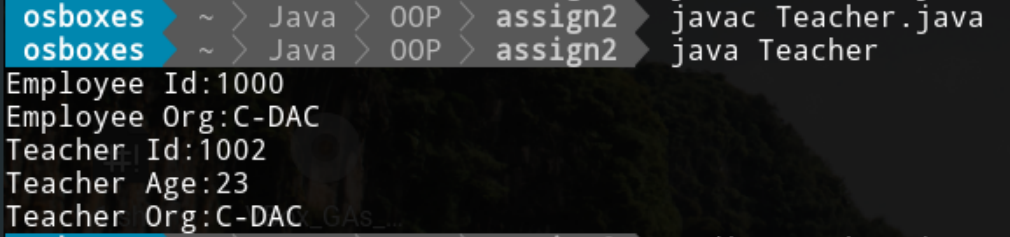
public static void main(String[]args){

Teacher t=new Teacher(1002,23); //object create of child class

t.display(); //call display method

}

}



**9. Write a program to make a request to invoke garbage collector**

class Simple{

public void finalize(){

System.out.println("Garbage Collector invoked!!!");

}

public static void main(String[]args){

Simple s=new Simple();

Simple s1=new Simple();

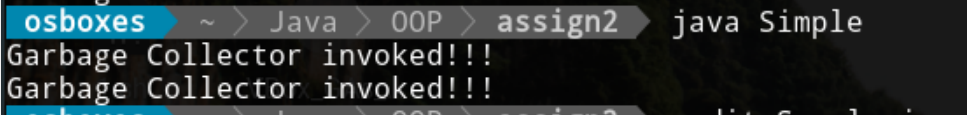
s=null; //unreferenced

s1=null;

System.gc(); //calling garbage collector method(finalize method automatic invoked)

}

}



**10. Write a program to demonstrate the use of nested class and its objects when nested class is a**

1. **private member of the outer class**

class Outer{

private int a=10; //private member

void display(){

class Inner{ //nested class

void msg()

{

System.out.println("Private data of Outer class:"+a);

}

}

Inner i=new Inner(); //nested class object

i.msg();

}

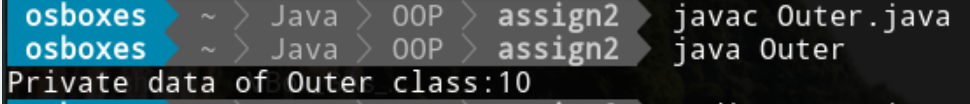
public static void main(String[]args){

Outer o=new Outer(); //outer class object

o.display();

}

}



1. **public member of the outer class.**

class Outer1{

public int a=10; //public member of outer class

public void disp(){

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

}

class Inner{

void msg(){

System.out.println("Public data of outer class:"+a); //access data of outer class

}

}

public static void main(String[]args)

{

Outer1 o=new Outer1(); //outer class object

Outer1.Inner i=o.new Inner(); //nested class object

i.msg();

o.disp();

}

}

