Object and Class in Java

In object-oriented programming technique, we design a program using objects and classes.

Object is the physical as well as logical entity whereas class is the logical entity only.

**Object in Java**

An entity that has state and behavior is known as an object e.g. chair, bike, marker, pen, table, car etc. It can be physical or logical (tengible and intengible). The example of integible object is banking system.

**An object has three characteristics:**

**state**: represents data (value) of an object.

**behavior**: represents the behavior (functionality) of an object such as deposit, withdraw etc.

**identity**: Object identity is typically implemented via a unique ID. The value of the ID is not visible to the external user. But,it is used internally by the JVM to identify each object uniquely.

**For Example**: Pen is an object. Its name is Reynolds, color is white etc. known as its state. It is used to write, so writing is its behavior.

Object is an instance of a class. Class is a template or blueprint from which objects are created. So object is the instance(result) of a class.

**Class in Java**

A class is a group of objects that has common properties. It is a template or blueprint from which objects are created.

**A class in java can contain:**

* data member
* method
* constructor
* block
* class and interface

**Syntax to declare a class:**

class<class\_name>{

data member;

method; }

**Example:**

class Student1{

int id;//data member (also instance variable)

String name;//data member(also instance variable)

public static void main(String args[]){

Student1 s1=new Student1();//creating an object of Student

System.out.println(s1.id);

System.out.println(s1.name); } }

**Output: 0 null**

**Instance variable in Java**

A variable that is created inside the class but outside the method, is known as instance variable.Instance variable doesn't get memory at compile time.It gets memory at runtime when object(instance) is created.That is why, it is known as instance variable.

**Method in Java**

In java, a method is like function i.e. used to expose behavior of an object.

**Advantage of Method**

* Code Reusability
* Code Optimization

**new keyword**

The new keyword is used to allocate memory at runtime.

**Example of Object and class that maintains the records of students**

In this example, we are creating the two objects of Student class and initializing the value to these objects by invoking the insertRecord method on it. Here, we are displaying the state (data) of the objects by invoking the displayInformation method.

class Student2{

introllno;

String name;

voidinsertRecord(int r, String n){ //method

rollno=r;

name=n;

}

void displayInformation()

{System.out.println(rollno+""+name);

}//method

public static void main(String args[]){

Student2 s1=new Student2();

Student2 s2=new Student2();

s1.insertRecord(111,"Karan");

s2.insertRecord(222,"Aryan");

s1.displayInformation();

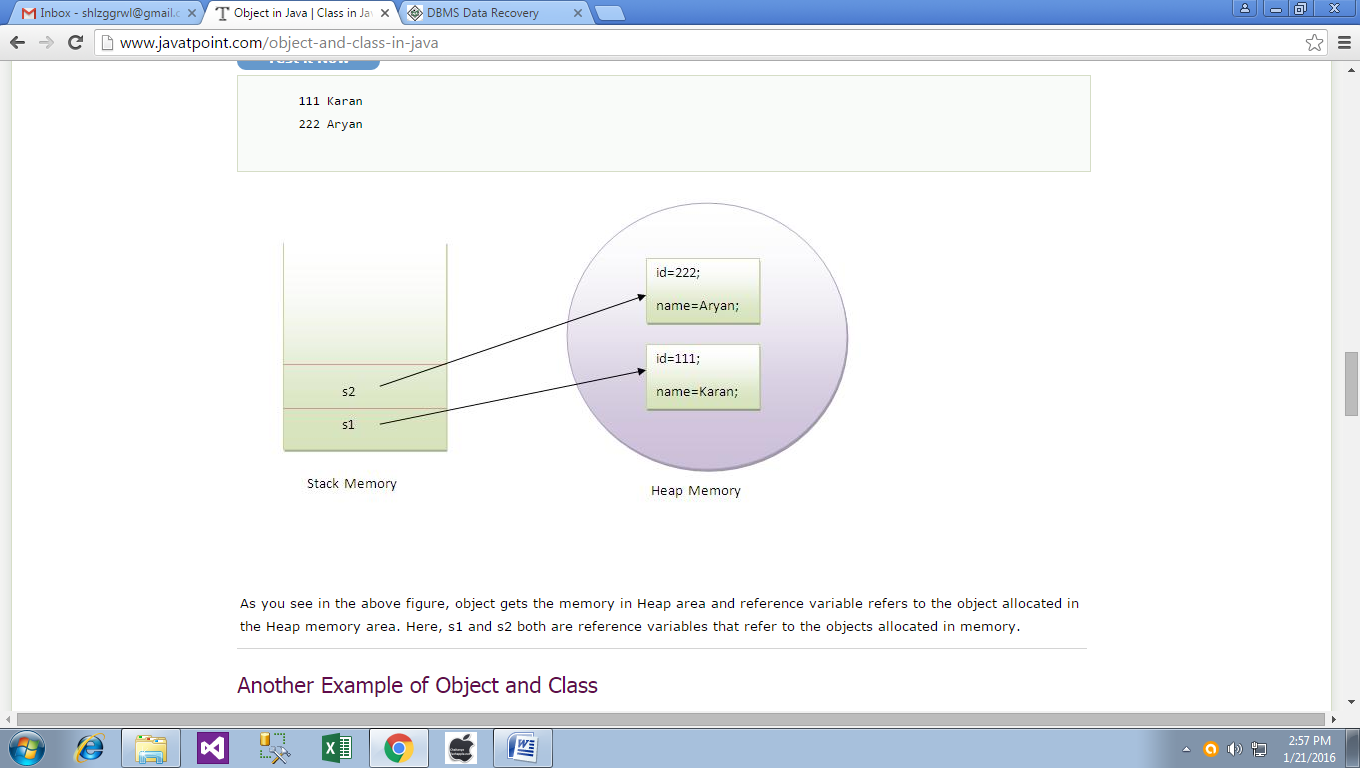
s2.displayInformation();

} }

**Output:**

111 Karan

222 Aryan



**Another Example of Object and Class**

There is given another example that maintains the records of Rectangle class. Its explanation is same as in the above Student class example.

class Rectangle{

int length;

int width;

void insert(intl,int w){

length=l;

width=w;

}

voidcalculateArea(){System.out.println(length\*width);}

public static void main(String args[]){

Rectangle r1=new Rectangle();

Rectangle r2=new Rectangle();

r1.insert(11,5);

r2.insert(3,15);

r1.calculateArea();

r2.calculateArea();

} }

**Output**:55

45

**Different ways to create an object in Java?**

There are many ways to create an object in java. They are:

* By new keyword
* By newInstance() method
* By clone() method
* By factory method etc.

**Anonymous object**

Anonymous simply means nameless.An object that have no reference is known as anonymous object.

If you have to use an object only once, anonymous object is a good approach.

class Calculation{

void fact(int n){

int fact=1;

for(inti=1;i<=n;i++){

fact=fact\*i;

}

System.out.println("factorial is "+fact);

}

public static void main(String args[]){

new Calculation().fact(5);//calling method with annonymous object

} }

**Output**:Factorial is 120

**Creating multiple objects by one type only**

We can create multiple objects by one type only as we do in case of primitives.

Rectangle r1=new Rectangle(),r2=new Rectangle();//creating two objects

**Example:**

class Rectangle{

int length;

int width;

void insert(intl,int w){

length=l;

width=w;

}

voidcalculateArea(){System.out.println(length\*width);}

public static void main(String args[]){

Rectangle r1=new Rectangle(),r2=new Rectangle();//creating two //objects

r1.insert(11,5);

r2.insert(3,15);

r1.calculateArea();

r2.calculateArea();

} }

**Output**:55

45