1. Object and Class in Java

Object in Java

An entity that has state and behavior is known as an object e.g. chair, bike, marker, pen, table, car etc.

- 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.

Class in Java

A class is a group of objects which have common properties. It is a template or blueprint from which objects are created. It is a logical entity. It can't be physical.

A class in Java can contain:

- **fields**
- **constructors**
- **blocks**
- **nested class and interface**

Syntax to declare a class:

```
1. class <class_name>{
2.     field;
3.     method;
4. }
```

Instance variable in Java

A variable which 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 run time 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 in Java

The new keyword is used to allocate memory at run time. All objects get memory in Heap memory area.

- Object and Class Example: main within class

In this example, we have created a Student class that have two data members id and name. We are creating the object of the Student class by new keyword and printing the objects value.

Here, we are creating main() method inside the class.

File: Student.java

```
1. class Student{
2.  int id;//field or data member or instance variable
3.  String name;
4.
5.  public static void main(String args[]){
6.   Student s1=new Student();//creating an object of Student
7.   System.out.println(s1.id);//accessing member through reference variable
8.   System.out.println(s1.name);
9.  }
10. }
```

Output:

```
0
null
```

2. Constructor in Java

Constructor in java is a *special type of method* that is used to initialize the object.

Java constructor is *invoked at the time of object creation*. It constructs the values i.e. provides data for the object that is why it is known as constructor.

- **Rules for creating java constructor**

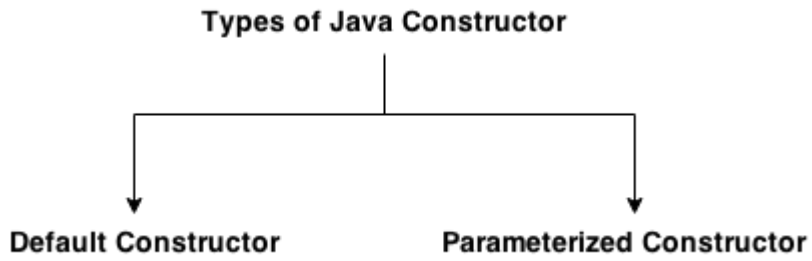
There are basically two rules defined for the constructor.

1. Constructor name must be same as its class name
2. Constructor must have no explicit return type

- **Types of java constructors**

There are two types of constructors:

1. Default constructor (no-arg constructor)
2. Parameterized constructor



Java Default Constructor

A constructor that have no parameter is known as default constructor.

- **Syntax of default constructor:**

1. <class_name>()
{
//Statements
}

- **Example of default constructor**

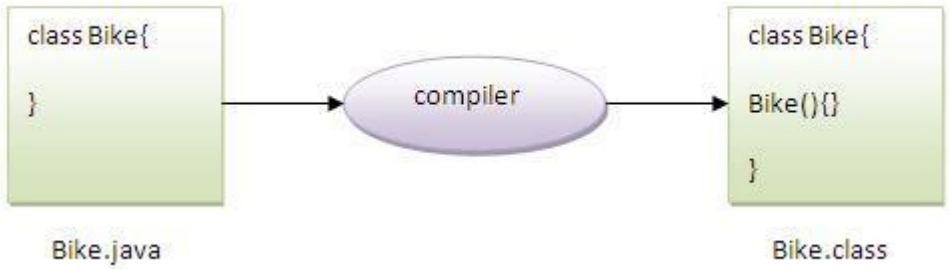
In this example, we are creating the no-arg constructor in the Bike class. It will be invoked at the time of object creation.

```
1. class Bike1{
2. Bike1(){System.out.println("Bike is created");}
3. public static void main(String args[]){
4. Bike1 b=new Bike1();
5. }
6. }
```

Output:

```
Bike is created
```

Rule: If there is no constructor in a class, compiler automatically creates a default constructor.



Q) **What is the purpose of default constructor?**

Default constructor provides the default values to the object like 0, null etc. depending on the type.

- **Example of default constructor that displays the default values**

```
1. class Student{
2. int id;
3. String name;
4.
5. void display(){System.out.println(id+" "+name);}
6.
7. public static void main(String args[]){
8. Student s1=new Student();
9. Student s2=new Student();
10. s1.display();
11. s2.display();
12. }
13. }
```

Output:

```
0 null
0 null
```

Explanation: In the above class,you are not creating any constructor so compiler provides you a default constructor.Here 0 and null values are provided by default constructor.

Java parameterized constructor

A constructor that have parameters is known as parameterized constructor.

Why use parameterized constructor?

Parameterized constructor is used to provide different values to the distinct objects.

- **Example of parameterized constructor**

In this example, we have created the constructor of Student class that have two parameters. We can have any number of parameters in the constructor.

```
1. class Student4{
2.     int id;
3.     String name;
4.
5.     Student4(int i,String n){
6.     id = i;
7.     name = n;
8.     }
9.     void display(){System.out.println(id+" "+name);}
10.
11.    public static void main(String args[]){
12.    Student4 s1 = new Student4(111,"Dhoni");
13.    Student4 s2 = new Student4(222,"Virat");
14.    s1.display();
15.    s2.display();
16.   }
17. }
```

Output:

Kalibermind Academy

```
111 Dhoni
222 Virat
```

Constructor Overloading in Java

Constructor overloading is a technique in Java in which a class can have any number of constructors that differ in parameter lists.The compiler differentiates these constructors by taking into account the number of parameters in the list and their type.

- **Example of Constructor Overloading**

```
1. class Student5{
2.     int id;
3.     String name;
4.     int age;
5.     Student5(int i,String n){
6.     id = i;
7.     name = n;
8.     }
9.     Student5(int i,String n,int a){
10.    id = i;
11.    name = n;
12.    age=a;
13.    }
14.    void display(){System.out.println(id+" "+name+" "+age);}
15.
16.    public static void main(String args[]){
17.    Student5 s1 = new Student5(111,"Karan");
18.    Student5 s2 = new Student5(222,"Aryan",25);
19.    s1.display();
20.    s2.display();
21.   }
22. }
```

Output:

```
111 Karan 0
222 Aryan 25
```

➢ Difference between constructor and method in java

There are many differences between constructors and methods. They are given below.

Java Constructor	Java Method
Constructor is used to initialize the state of an object.	Method is used to expose behaviour of an object.
Constructor must not have return type.	Method must have return type.
Constructor is invoked implicitly.	Method is invoked explicitly.
The java compiler provides a default constructor if you don't have any constructor.	Method is not provided by compiler in any case.
Constructor name must be same as the class name.	Method name may or may not be same as class name.