Basics on Java Programming

When we consider a Java program it can be defined as a collection of objects that communicate via invoking each other's methods.

- **Object -** Objects have states and behaviors. Example: A dog has states - color, name, breed as well as behaviors -wagging, barking, eating. An object is an instance of a class.
- **Class -** A class can be defined as a template/ blue print that describes the behaviors/states that object of its type support.
- **Methods -** A method is basically a behavior. A class can contain many methods. It is in methods where the logics are written, data is manipulated and all the actions are executed.
- **Instance Variables -** Each object has its unique set of instance variables. An object's state is created by the values assigned to these instance variables.

Basics on Java Programming

Basic Syntax:

- **Case Sensitivity -** Java is case sensitive, which means identifier **Hello** and **hello** would have different meaning in Java.
- **Class Names -** For all class names the first letter should be in Upper Case.
- **Method Names -** All method names should start with a Lower Case letter.
- **Program File Name -** Name of the program file should exactly match the class name.
- **public static void main(String args[]) -** Java program processing starts from the main() method which is a mandatory part of every Java program.

Access Specifiers

Java Access Specifiers (also known as Visibility Specifiers) regulate access to classes, fields and methods in Java. These Specifiers determine whether a field or method in a class, can be used or invoked by another method in another class or sub-class.

There are four types of Java access modifiers:

- **Private**: The access level of a private modifier is only within the class. It cannot be accessed from outside the class.
- **Default**: The access level of a default modifier is only within the package. It cannot be accessed from outside the package. If you do not specify any access level, it will be the default.
- **Protected**: The access level of a protected modifier is within the package and outside the package through child class. If you do not make the child class, it cannot be accessed from outside the package.
- **Public**: The access level of a public modifier is everywhere. It can be accessed from within the class, outside the class, within the package and outside the package.

Continued…

Access Modifier	within class	within package	outside package by subclass only	outside package
Private	Y	N	N	N
Default	Y	Y	N	N
Protected	Y	Y	Y	N
Public	Y	Y	Y	Y

Java Constructor

Constructor

- A **constructor in Java** is a block of code similar to a method that's called when an instance of an object is created.
- Here are the key differences between a **constructor** and a method: A **constructor** doesn't have a return type.
- The name of the **constructor** must be the same as the name of the class.

Rules for creating java constructor

There are basically two rules defined for the constructor.

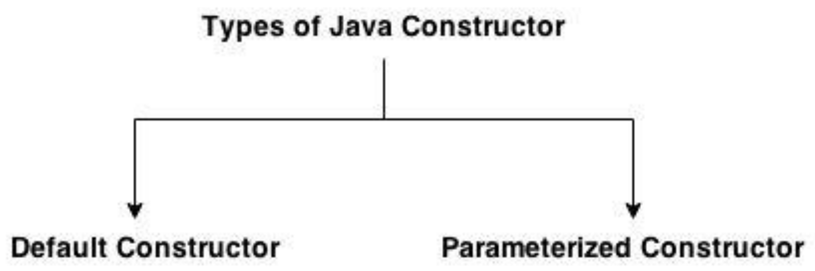
- Constructor name must be same as its class name
- Constructor must have no explicit return type

The constructor gets called when we create an object of a class

Types of java constructors

There are two types of constructors:

- Default constructor (no-arg constructor)
- Parameterized constructor



Java Default Constructor

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

Syntax of default constructor:

```
<class_name>()
{
        …………..
}
```

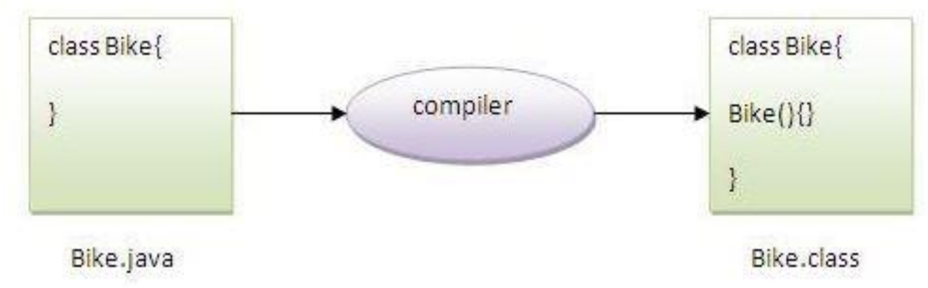
Example of default constructor

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

default constructor

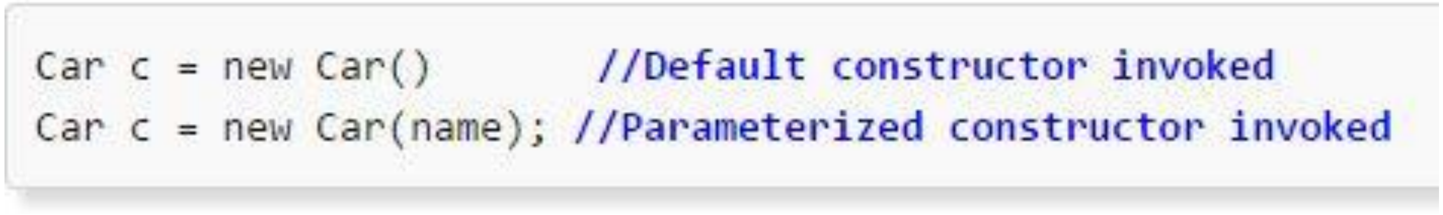
Rule: If there is no constructor in a class, compiler automatically creates a default constructor.

- Defa default valu tc. depe



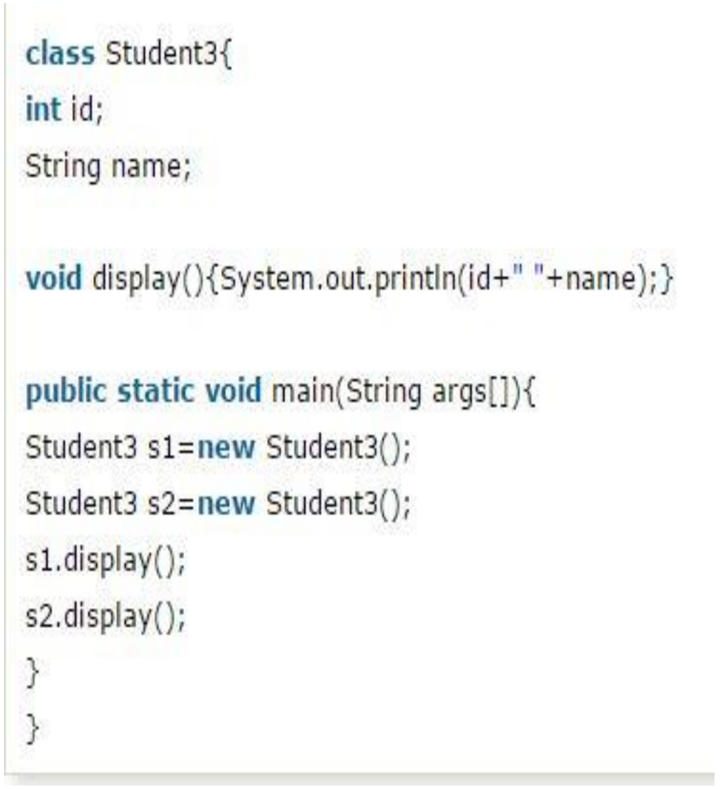
Java parameterized constructor

- A constructor that have parameters is known as parameterized constructor.
- Parameterized constructor is used to provide different values to the distinct objects.

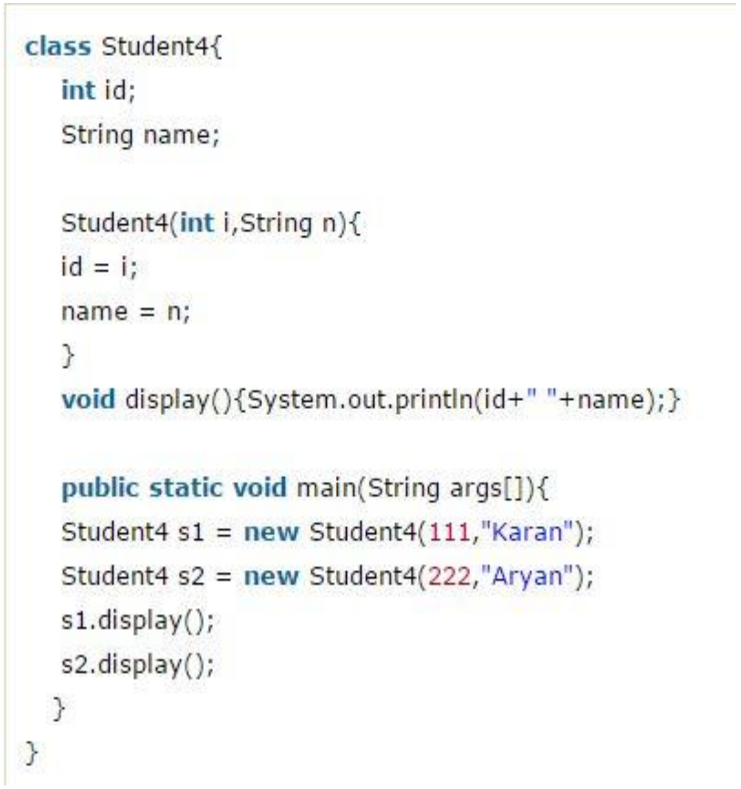


```
Car c = new Car()       //Default constructor invoked
Car c = new Car(name); //Parameterized constructor invoked
```

Example



```
class Student3{
int id;
String name;

void display(){System.out.println(id+" "+name);}

public static void main(String args[]){
Student3 s1=new Student3();
Student3 s2=new Student3();
s1.display();
s2.display();
}
}
```

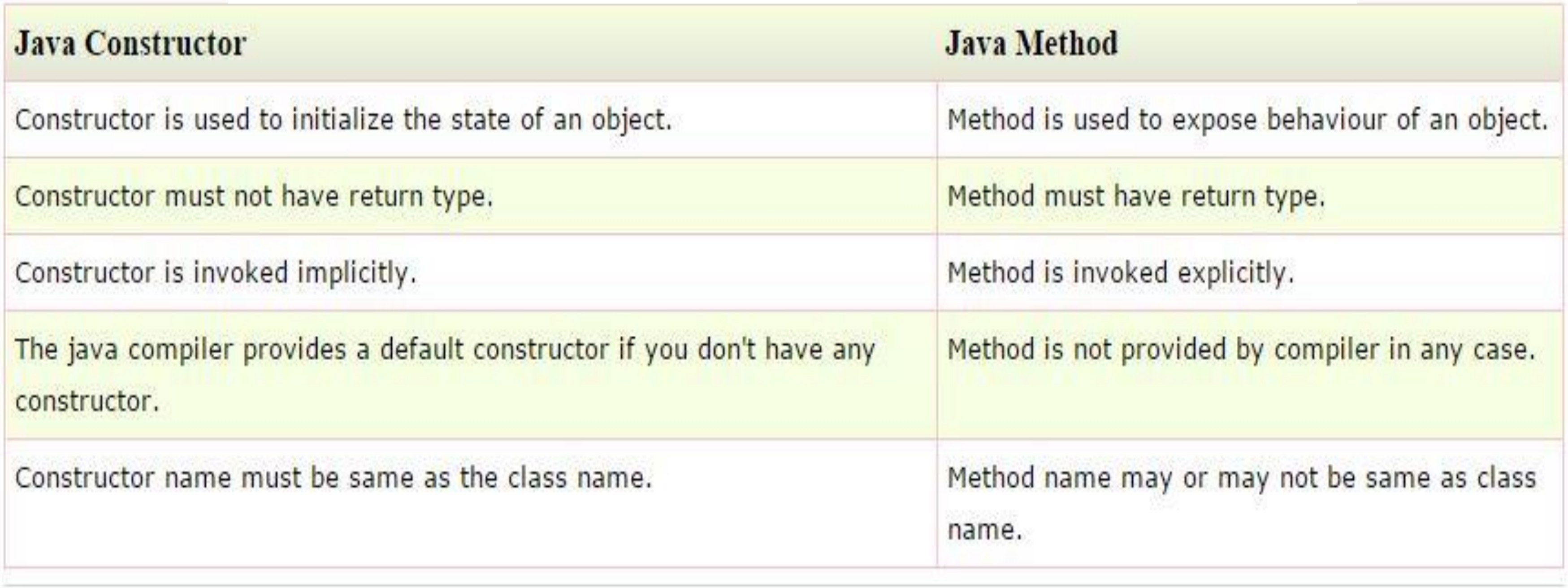


```
class Student4{
    int id;
    String name;

    Student4(int i,String n){
    id = i;
    name = n;
    }
    void display(){System.out.println(id+" "+name);}

    public static void main(String args[]){
    Student4 s1 = new Student4(111,"Karan");
    Student4 s2 = new Student4(222,"Aryan");
    s1.display();
    s2.display();
   }
}
```

Constructors

- ✓ A constructor initializes an object immediately upon creation.
- ✓ It is similar to a method and has same name as the class in which it resides.
- ✓ It has no return type not even void.
- ✓ Once defined it is automatically called immediately after the object is created.
- ✓ When there is no explicit constructor for a class, then java creates a default constructor for that class.
- ✓ The default constructor automatically initializes all instance variables to 0.

Difference between constructor and method



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.