UNIT III

OBJECT AND CLASSES

CONSTRUCTORS

InJava, a constructor is a block of codes similar to the method. It is called when an instance of the classis created. At the time of calling constructor, memory for the object is allocated in the memory.

Itisaspecialtypeof methodwhichisusedto initializethe object.

Every time an object is created using the new() keyword, at least one constructor is called.

It calls a default constructor if there is no constructor available in the class. In such case, Java compiler provides a default constructor by default.

There are two types of constructors in Java: no-arg constructor, and parameterized constructor.

Note: It is called constructor because it constructs the values at the time of object creation. It is not necessary to write a constructor for a class. It is because java compiler creates a default constructor if your class doesn't have any.

RulesforcreatingJavaconstructor

Therearetworulesdefinedfortheconstructor.

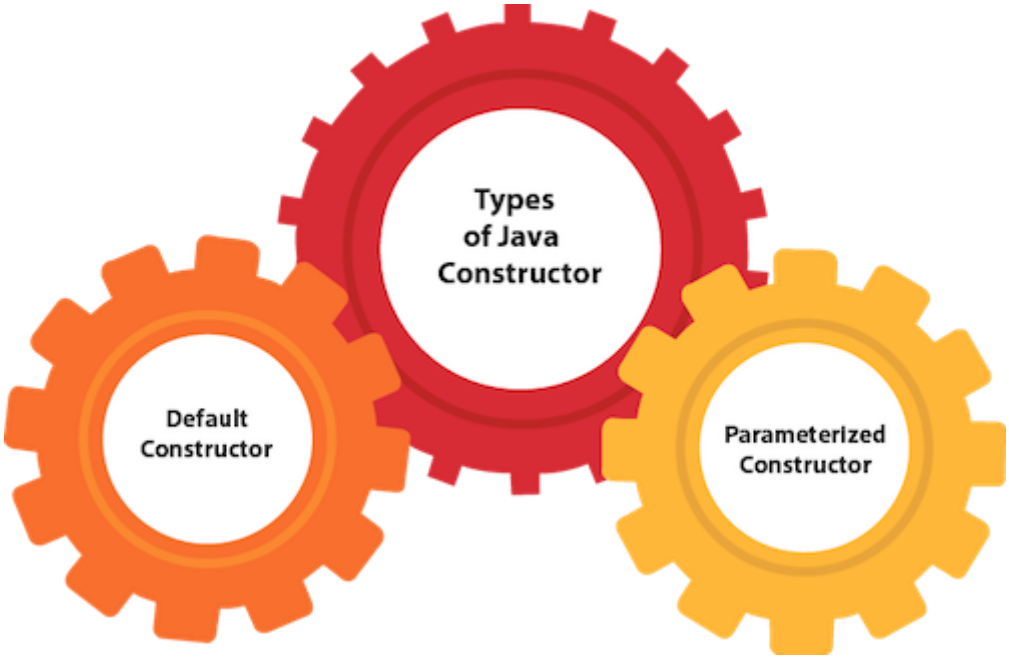
1. Constructornamemustbethesameasitsclassname
2. AConstructormusthavenoexplicitreturntype
3. AJavaconstructorcannotbeabstract,static,final,andsynchronized

> Note:Wecanuseaccessmodifierswhiledeclaringaconstructor.It controls the object creation. In other words, we can have private, protected, public or default constructor in Java.

TypesofJavaconstructors

TherearetwotypesofconstructorsinJava:

1. Defaultconstructor(no-argconstructor)
2. Parameterizedconstructor



JavaDefaultConstructor

A constructor is called "Default Constructor" when it doesn't have anyparameter.

Syntaxofdefaultconstructor:

<class_name>(){}

Exampleofdefaultconstructor

```
//JavaProgramtocreateandcalladefaultconstructor class
Bike1{
//creatingadefaultconstructor
```

```
Bike1(){System.out.println("Bikeiscreated");}
//mainmethod
publicstaticvoidmain(Stringargs[]){
//callingadefaultconstructor Bike1
b=new Bike1();
}
}
```

Testit Now

Output:

Bikeiscreated

Rule:Ifthereisnoconstructorinaclass, compilerautomatically createsadefaultconstructor.

Q)Whatisthepurposeof adefaultconstructor?

Thedefaultconstructorisusedtoprovidethedefaultvaluestothe object like 0, null, etc., depending on the type.

Exampleofdefaultconstructorthatdisplaysthe default values

```
//Letusseeanotherexampleofdefaultconstructor
//whichdisplaysthedefaultvalues
classStudent3{
intid;
```

```
Stringname;
//method todisplaythevalueof idand name
voiddisplay(){System.out.println(id+""+name);}

publicstaticvoidmain(Stringargs[]){
//creatingobjects
Student3s1=newStudent3();
Student3s2=new Student3();
//displayingvaluesoftheobject
s1.display();
s2.display();
}
}
```

Output:

```
0null
0null
```

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.

JavaParameterizedConstructor

Aconstructorwhichhasaspecificnumberofparametersiscalleda parameterized constructor.

Whyusetheparameterizedconstructor?

Theparameterizedconstructorisusedtoprovidedifferentvaluestodistinct objects. However, you can provide the same values also.

Exampleofparameterizedconstructor

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.

```
//JavaProgramtodemonstratetheuseoftheparameterizedconstructor
.
classStudent4{
    intid;String
    name;
    //creatingaparameterizedconstructorStudent4(in
    ti,String n){
    id = i;
    name=n;
    }
    //methodtodisplaythevalues
    voiddisplay(){System.out.println(id+""+name);}

    publicstaticvoidmain(Stringargs[]){
    //creating objects and passing values
    Student4s1=new Student4(111,"Karan");
    Student4s2 =newStudent4(222,"Aryan");
    //callingmethodtodisplaythevaluesofobject
    s1.display();
    s2.display();
   }
}
```

Output:

```
111 Karan
222Aryan
```

ConstructorOverloadinginJava

InJava,aconstructorisjust likea methodbut without returntype. It can also be overloaded like Java methods.

Constructoroverloading in Javais a technique of having more than one constructor with different parameter lists. They are arranged in a way that each constructor performs a different task. They are differentiated by the compiler by the number of parameters in the list and their types.

ExampleofConstructorOverloading

```
//Javaprogramtooverload constructors
classStudent5{
   intid;String
   name;
   intage;
   //creatingtwoargconstructorStu
   dent5(inti,String n){
   id = i;
   name=n;
   }
   //creatingthreeargconstructorSt
   udent5(inti,Stringn,inta){ id =
   i;
   name=n;
   age=a;
   }
   voiddisplay(){System.out.println(id+""+name+""+age);}

   public static void main(String args[]){
   Student5 s1 = new Student5(111,"Karan");
   Student5s2=newStudent5(222,"Aryan",25);
   s1.display();
   s2.display();
  }}
```

Output:

111Karan0
222Aryan25

Differencebetweenconstructorandmethodin Java

JavaConstructor	JavaMethod
Aconstructorisusedtoinitialize the state of an object.	Amethodisusedtoexposethe behavior of an object.
Aconstructormustnothavea return type.	Amethodmusthaveareturn type.
Theconstructorisinvoked implicitly.	Themethodisinvoked explicitly.
The Java compiler provides a default constructorif you don't haveanyconstructorinaclass.	Themethodisnotprovidedby the compiler in any case.
Theconstructornamemustbe same as the class name.	Themethodnamemayormay notbesameastheclass name.

JavaCopyConstructor

There is no copy constructor in Java. However, we can copy the values from one object to another like copy constructor in C++.

Therearemanywaystocopythevaluesofoneobjectintoanotherin Java. They are:

- Byconstructor
- Byassigningthevalues ofoneobjectintoanother
- Byclone()methodofObjectclass

Inthisexample,wearegoingtocopythevaluesofoneobjectintoanother using Java constructor.

```
//Javaprogramtoinitializethevaluesfromoneobjecttoanotherobject.
class Student6{
    int id;String
    name;
    //constructortoinitializeintegerandstringStudent
    6(int i,String n){
    id = i;
    name=n;
    }
    //constructortoinitializeanotherobject
    Student6(Student6 s){
    id =s.id;
    name =s.name;
    }
    void display(){System.out.println(id+""+name);}

    public static void main(String args[]){
    Student6 s1=new Student6(111,"Karan");
    Student6 s2 = new Student6(s1);
    s1.display();
    s2.display();
```

}}
Output:

111 Karan
111Karan

Copyingvalueswithoutconstructor

Wecancopythevaluesofoneobjectintoanotherbyassigningtheobjects values to another object.

```
classStudent7{
  intid;String name;
  Student7(inti,Stringn){ id
  = i;name = n;
  }
  Student7(){}
  voiddisplay(){System.out.println(id+""+name);}
  public static void main(String args[]){
  Student7s1=newStudent7(111,"Karan");
  Student7 s2 = new Student7();
  s2.id=s1.id;
  s2.name=s1.name;
  s1.display();   s2.display();
 }}
```

Output:

111 Karan

111 Karan