Sanaka Educational Trust's Group Of Institutions- [SETGOI]

Name : Arnab Dutta
Roll No. : 27800120035

Sub-Code : PCC-CS503
Subject : Object Oriented Programming
Topic : Constructors in Java
Date:29/08/2022

Constructors in Java

In Java, a constructor is a block of codes similar to the method. It is called when an instance of the class is created. At the time of calling constructor, memory for the object is allocated in the memory.
It is a special type of method which is used to initialize the 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.

Now let us come up with the syntax for the constructor being invoked at the time of object or instance creation.

```
class Geek
{
  .......

  // A Constructor
  new Geek() {
  }

  .......
}

// We can create an object of the above class
// using the below statement. This statement
// calls above constructor.
Geek obj = new Geek();
```

The first line of a constructor is a call to super() or this(), (a call to a constructor of a super-class or an overloaded constructor), if you don't type in the call to super in your constructor the compiler will provide you with a non-argument call to super at the first line of your code, the super constructor must be called to create an object:

```
import java.io.*;

class Geeks {
    Geeks() { super(); }
    public static void main(String[] args)
    {
        Geeks geek = new Geeks();
    }
}
```

- **Need of Constructor :**

constructors are used to assign values to the class variables at the time of object creation, either explicitly done by the programmer or by Java itself (default constructor).

- **When is a Constructor called?**

Each time an object is created using a new() keyword, at least one constructor (it could be the default constructor) is invoked to assign initial values to the data members of the same class.
Characteristics of Constructors

- **Characteristics of Java Constructors :**

- Constructors name must be similar to that of the class name inside which it resides.
- Constructors cannot be private.
- A constructor can be overloaded.
- Constructors cannot return a value.
- Constructors do not have a return type; not even void.
- An abstract class can have the constructor.
- Constructors are automatically called when an object is created.

Types of Constructor :-

In Java, constructors can be divided into 3 types: 1. No-Arg Constructor 2. Parameterized Constructor 3. Default Constructor

• No-Arg Constructor :-

Similar to methods, a Java constructor may or may not have any parameters (arguments).If a constructor does not accept any parameters, it is known as a no-argument constructor. For example,

```
class Company {
String name;
// public constructor
public Company() {
name = " SETGOI";
}
 }
class Main {
 public static void main(String[] args) {
// object is created in another class
Company obj = new college ();
 System.out.println("college name = " + obj.name);
}
 }
```

Output: College name = SETGOI

• Parameterized Constructor :-

A Java constructor can also accept one or more parameters. Such constructors are known as parameterized constructors (constructor with parameters).

```
class Main {
String languages;
// constructor accepting single value
Main(String lang) {
languages = lang;
System.out.println(languages + " Programming Language");
}
public static void main(String[] args) {
// call constructor by passing a single value
Main obj1 = new Main("Java");
Main obj2 = new Main("Python");
Main obj3 = new Main("C");
}
}
```

Output: Java Programming Language
Python Programming Language
C Programming Language

Default Constructor :-

If we do not create any constructor, the Java compiler automatically create a no-arg constructor during the execution of the program. This constructor is called default constructor

```
//Java Program to create and call a default constructor
class Bike1{
//creating a default constructor
Bike1(){System.out.println("Bike is created");}
//main method
public static void main(String args[]){
//calling a default constructor
Bike1 b=new Bike1();
}
}
```

Output :

Bike is created

Important Notes on Java Constructors

• Constructors are invoked implicitly when you instantiate objects.

• The two rules for creating a constructor are: The name of the constructor should be the same as the class. A Java constructor must not have a return type

.• If a class doesn't have a constructor, the Java compiler automatically creates a default constructor during run-time. The default constructor initializes instance variables with default values. For example, the int variable will be initialized to 0

• Constructor types: No-Arg Constructor - a constructor that does not accept any arguments Parameterized constructor - a constructor that accepts arguments Default Constructor - a constructor that is automatically created by the Java compiler if it is not explicitly defined.

• A constructor cannot be abstract or static or final.

• A constructor can be overloaded but can not be overridden

Reference :

1. https://www.programiz.com/
2. https://www.geeksforgeeks.org/
3. https://www.javatpoint.com/