Course Name - Object Oriented Programming using Java

Lecture 10 – Constructor- Definition, Usage of Constructor, Different types of Constructor.

Presented By
Dr. Sudipta Sahana
Asso. Prof.
Dept. of CSE
UEM - Kolkata

Topic of Interest

► Constructor

► Types of java constructors

► Java Default Constructor

► Java parameterized constructor

► Java Copy Constructor

► When constructor is actually called

► Difference between constructor and method in java

Constructor

In Java, constructor is a block of codes similar to method. It is called when an instance of object is created and memory is allocated for the object.

When a constructor is called:
Every time an object is created using new() keyword, at least one constructor is called. It is called a default constructor.

Following is the syntax of a constructor

class A

{ A(){

* System.out.println("From contstructor"); }*

public static void main(String[] args){

* A a=new A();}}*

Output: From Constructor

Types of java constructors:

- Constructors are roughly of 2 types… No-argument constructors and argument-constructors.

- Caution: No-argument constructors are not necessarily the default constructor. Default constructor is the no-argument constructor provided by the compiler should the developer give none.

- Once you declare at least one argument-constructor inside your class, your no-argument default constructor would not be available at all until you yourself provide your no-argument constructor.

- Copy constructor is a special type of argument constructor where the argument is an object of the same class.

Java Default Constructor:

A constructor is called "Default Constructor" when the constructor is not defined in code but it is called during object creation.

Example of default constructor constructor

```
class Bike{
void colour(){System.out.
println("The bike is Red");}

public static void main(String
args[]){
Bike b=new Bike();
b.colour();
}  }

Output: The bike is Red
```

Example of no argument

```
class A
{ A(){
 System.out.println("From
contstructor"); }
public static void main(String[] args){
 A a=new A();}}

Output: From Constructor
```

Java parameterized constructor:

A constructor which has a specific number of parameters is called parameterized constructor. Parameterized constructor is used to provide different values to the distinct objects.

Example of parameterized constructor

```java
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();
    }
}
```

Output

111 Karan
222 Aryan

Java Copy Constructor:

There is no copy constructor in java. But, we can copy the values of one object to another like copy constructor in C++.

class A
{
int x;
A(A objRef){
this.x=objRef.x;
}
}

It's a copy constructor. It takes as argument an object of the same class and initializes the instance value accordingly.
Since it can perform the job of copying data members, it's named "Copy"

when a constructor is actually called

- In a class, all the static blocks and initializers are executed first once and only once in the order in which they appear in the code, when the class is loaded into the memory.

- Then non-static blocks, often known as instance blocks and initializers are executed in the order in which they appear. They are execute for each object creation. Instance fields are allocated into memory with corresponding default values.

- Then constructor is called and inside the constructor the call to super(<>) must be the first statement. Well, more on *super* later

Difference between constructor and method in java:

CONSTRUCTORS	METHODS
A Constructor is a block of code that initializes a newly created object.	A Method is a collection of statements which returns a value upon its execution.
A Constructor is invoked when a object is created using the keyword new.	A Method is invoked through method calls.
It has no return type not even void	A Method must have a return type.
A Constructor initializes a object that doesn't exist.	A Method does operations on an already created object.
A Constructor's name must be same as the name of the class.	A Method's name can be anything.
A Constructor cannot be inherited by subclasses.	A Method can be inherited by subclasses.
Constructor can't be static.	Method can be static.