**Java – Default constructor with example**

Earlier we discussed [**constructor overloading**](http://beginnersbook.com/2013/05/constructor-overloading/) and [**constructor chaining**](http://beginnersbook.com/2013/12/java-constructor-chaining-with-example/). In this tutorial we will discuss few things about default constructor. A **[constructor](http://beginnersbook.com/2013/03/constructors-in-java/" \t "_blank)** with no argument is known as **default constructor** in java.

Lets see a simple example first: Refer the comments in the below program.

class NoteBook{

/\*This is my default constructor. A constructor does

\* **not** have a return type and it's name

\* should exactly same as class name

\*/

NoteBook(){

System.out.println("Default constructor");

}

public void mymethod()

{

System.out.println("Void method of the class");

}

public static void main(String args[]){

/\* Creating object of class using default constructor

\* (new NoteBook()) so default constructor would be

\* invoked

\*/

NoteBook obj = new NoteBook();

obj.mymethod();

}

}

Output:

Default constructor

Void method of the class

As you can see in the above example that for creating the object of the class we have used the statement: NoteBook obj = new NoteBook();. HereNoteBook is class name, obj is object of class NoteBook, new keyword is for creating a new object and NoteBook() is the default constructor.

**Note**: If you try to create an object like this in above program: NoteBook obj = new NoteBook(12); then it would throw a compilation error becauseNoteBook(12) is referring to a constructor with single int argument, since we didn’t have a constructor with int argument in above example. The program would throw a compilation error in this case.

However the same does not apply for default constructor: Even if you do not define a default constructor in the class, the compiler does that for you implicitly.

**Consider the below example**: Here I didn’t declare any **default constructor**and I created the object of class using default constructor (new Example()) even then program ran fine without any issues. This shows that compiler creates a default empty constructor for a class if there is no constructor defined in it.

class Example{

//I did not define any constructor here

public void disp()

{

System.out.println("disp method of Example class");

}

public static void main(String args[]){

Example obj2 = new Example();

obj2.disp();

}

}

Output:

disp method of Example class