Lesson 3.2

Static Block

By the end of this lesson you will learn:

- What is Static Block
- How Static Block works
- Why Static Block is Called Before Main Method
- Why We Need Static Block
- Multiple Static Block

What is Static Block

A static block in Java is a block of code that is executed at the time of loading a class for use in a Java application. Static block is mainly used for the static initializations of a class. A static block gets executed exactly **once** when the class is first loaded.

A static initialization block is a normal block of code enclosed in braces, **{ }**, and preceded by the **static** keyword. Here is an example:

```
static {
    // whatever code is needed for initialization goes here
}
```

How Static Block Works

Let us examine the following example:

```java
class Test {
    static int i;

    static {
        i = 10;
        System.out.println("static block called ");
    }
}

class Main {
    public static void main(String args[]) {
        System.out.println(Test.i);
    }
}
```

Output:

```
static block called
10
```

1. In the above example, the static variable i, has been initialized with a value10 in the static initializer block.
2. Although we **do not have an object of Test**, static block is called because i is being accessed in following statement.
3. Static Block is called when the class loads in JVM. That is why static block does not depend on constructor. Even static block is called before an object. See the below example.
4. Although we have **two objects** of Test class in the below example, static block is executed only once.

```java
class Test {
    static int i;

    static {
        i = 10;
        System.out.println("static block called ");
    }
    Test(){
        System.out.println("Constructor called");
    }
}

class Main {
    public static void main(String args[]) {
       Test t1 = new Test();
       Test t2 = new Test();
    }
}
```

```
Output:
static block called
Constructor called
Constructor called
```

Why Static Block is Called Before Main Method

When we execute a class, JVM performs two actions at the runtime.

1. JVM loads corresponding dot (.) class file (byte code) into memory.
2. During the dot (.) class file loading into memory, the static blocks are executed. After the loading dot class file, JVM calls the main method to start execution. Therefore, static block is executed before the main method.

In other words, we can also say that Static block always gets executed first in Java because they are stored in the memory at the time of class loading or before the object creation.

Why We Need Static Block

1. The most powerful use of a static block can be realized while performing operations that are required to be executed only once for a Class in an application lifecycle. So, if we want or need to execute some code for every object, we can use static initializer block.
2. They are used to initialize the static variables.

Multiple Static Block

A class can have multiple static initialization blocks that will execute in the same sequence as written in the program. i.e. The order of execution of multiple static initialization blocks is executed automatically from top to bottom during the dot class file loading.

Let us take an example:

```
public class MultipleStaticBlocks {
      static {
            System.out.println("Static block-1");
      }
      static {
            System.out.println("Static block-2");
      }
      public static void main(String[] args) {
            new MultipleStaticBlocks();
      }
}
```

```
Output:
Static block-1
Static block-2
```

Exercise

1. Take the first example from “How static block works” section. Write the program in your computer and try to execute it.
2. Now remove this “`System.out.println(Test.i);`” line and then monitor the output.
3. Later, create an empty main method in the same Test class. Execute it and compare the results with the previous question.