

Beginners Book: Java Basics

Lesson 4: Variables In Java

Declaring Variables In Java:

To declare a variable follow this syntax:

```
data_type variable_name = value;
```

In the above syntax the value is optional because in Java you are able to declare a variable and assign a value at a later time.

Example:

```
int num;
```

Here num is a variable and int is a data type

You can also assign the values to the variables while declaring them such as below.

```
char ch = 'A';  
int number = 100;
```

OR like this:

```
char ch;  
int number;  
ch = 'A';  
number = 100;
```

Java Variable Naming Convention:

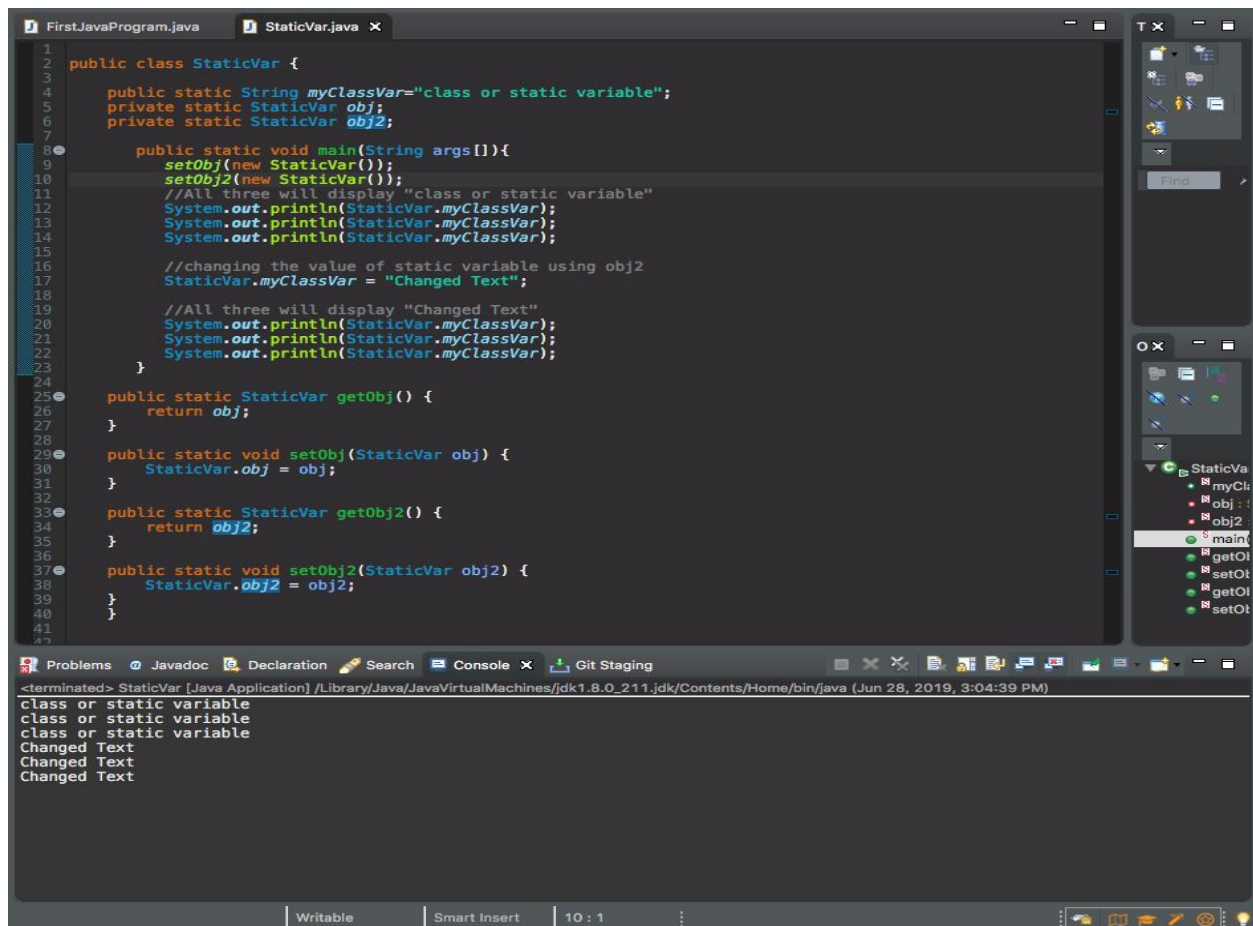
- Variable naming cannot contain white spaces, i.e. `int num ber = 100;` would be invalid because the variable name has space in it
- Variable names can begin with special characters such as \$ and _
- Variable names should begin with a lowercase letter, for example `int number;`
- For lengthy variables names that have more than one word the first letter in the first word should be lowercase and the rest of the words should be uppercase i.e. `int smallNumber;`
- Variable names are case sensitive in Java

Types of Variables in Java

Static (or class) Variable

Static variables are also known as class variable because they are associated with the class and common for all the instances of class.

Example of Static Variable



```
1 public class StaticVar {
2
3     public static String myClassVar="class or static variable";
4     private static StaticVar obj;
5     private static StaticVar obj2;
6
7
8     public static void main(String args[]){
9         setObj(new StaticVar());
10        setObj2(new StaticVar());
11        //All three will display "class or static variable"
12        System.out.println(StaticVar.myClassVar);
13        System.out.println(StaticVar.myClassVar);
14        System.out.println(StaticVar.myClassVar);
15
16        //changing the value of static variable using obj2
17        StaticVar.myClassVar = "Changed Text";
18
19        //All three will display "Changed Text"
20        System.out.println(StaticVar.myClassVar);
21        System.out.println(StaticVar.myClassVar);
22        System.out.println(StaticVar.myClassVar);
23    }
24
25    public static StaticVar getObj() {
26        return obj;
27    }
28
29    public static void setObj(StaticVar obj) {
30        StaticVar.obj = obj;
31    }
32
33    public static StaticVar getObj2() {
34        return obj2;
35    }
36
37    public static void setObj2(StaticVar obj2) {
38        StaticVar.obj2 = obj2;
39    }
40
41 }
```

StaticVa
myCl
obj :
obj2
main
getO
setO
getO
setO

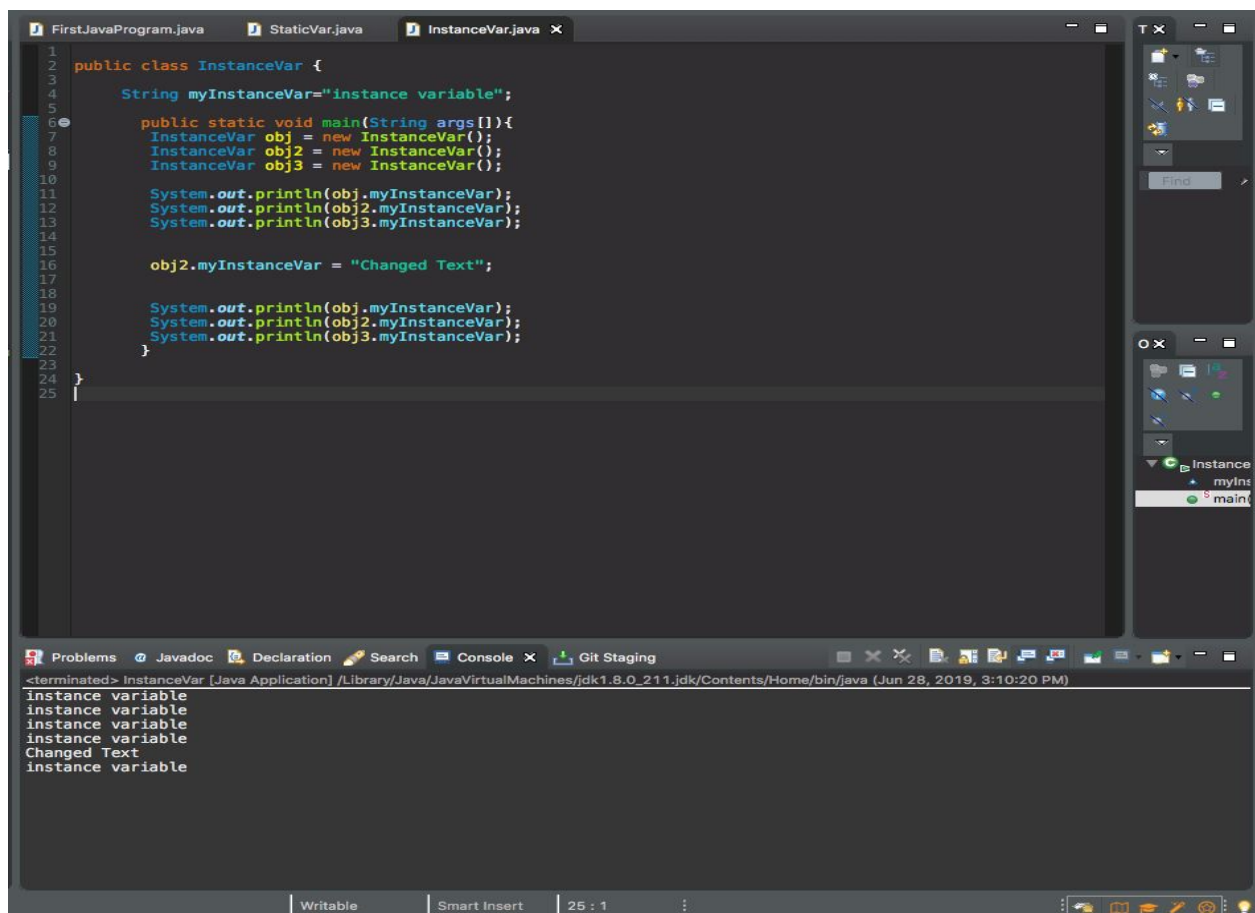
<terminated> StaticVar [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/bin/java (Jun 28, 2019, 3:04:39 PM)
class or static variable
class or static variable
class or static variable
Changed Text
Changed Text

Writable Smart Insert 10 : 1

Instance Variable

Each instance(objects) of class has its own copy of instance variable. Unlike static variable, instance variables have their own separate copy of instance variable. We have changed the instance variable value using object obj2 in the following program and when we displayed the variable using all three objects, only the obj2 value got changed, others remain unchanged. This shows that they have their own copy of instance variable.

Example of Instance Variable



```
1 public class InstanceVar {
2
3     String myInstanceVar="instance variable";
4
5     public static void main(String args[]){
6         InstanceVar obj = new InstanceVar();
7         InstanceVar obj2 = new InstanceVar();
8         InstanceVar obj3 = new InstanceVar();
9
10        System.out.println(obj.myInstanceVar);
11        System.out.println(obj2.myInstanceVar);
12        System.out.println(obj3.myInstanceVar);
13
14        obj2.myInstanceVar = "Changed Text";
15
16        System.out.println(obj.myInstanceVar);
17        System.out.println(obj2.myInstanceVar);
18        System.out.println(obj3.myInstanceVar);
19    }
20 }
21
22
23
24
25
```

Problems Javadoc Declaration Search Console X Git Staging

<terminated> InstanceVar [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/bin/java (Jun 28, 2019, 3:10:20 PM)

```
instance variable
instance variable
instance variable
instance variable
Changed Text
instance variable
```

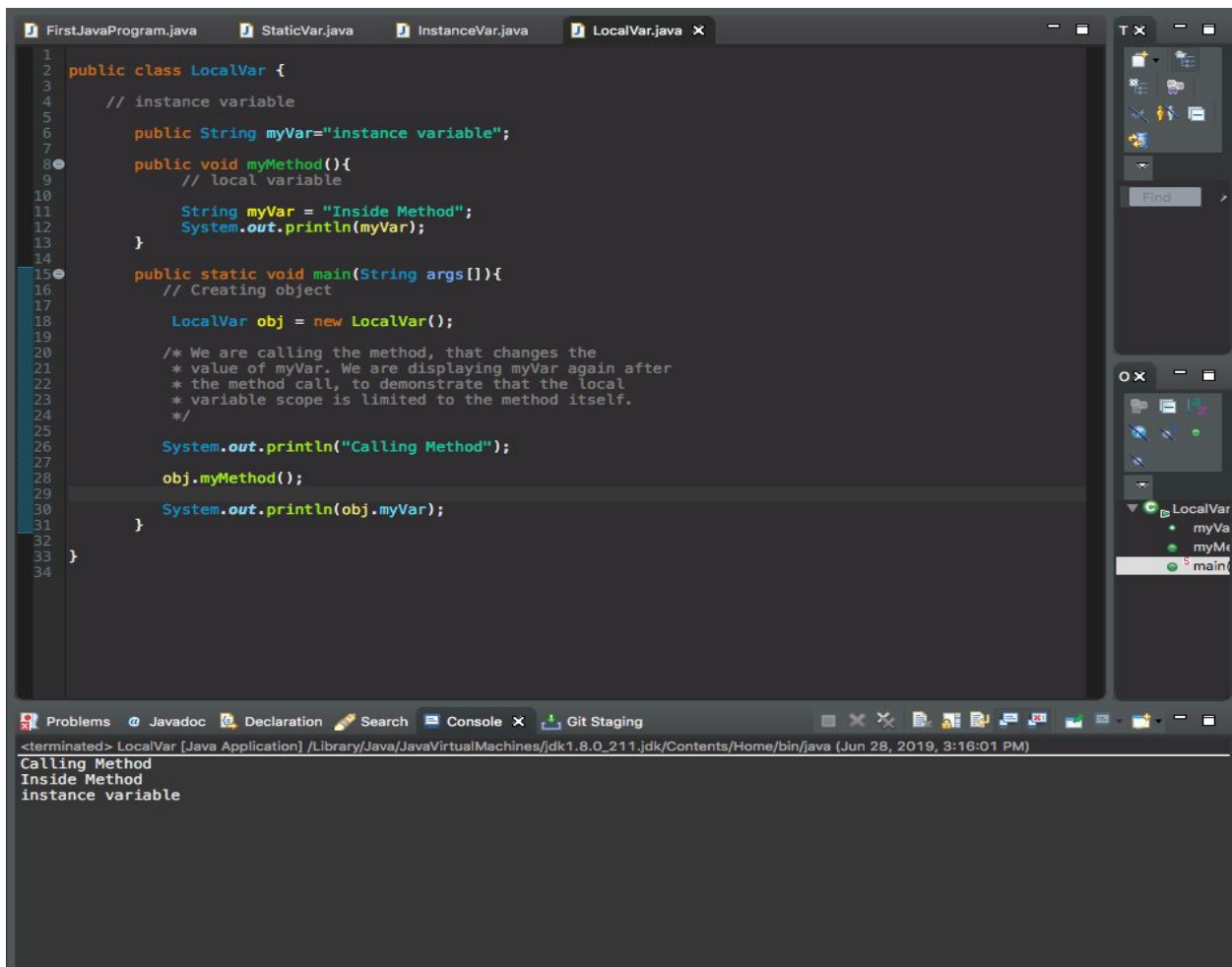
Writable Smart Insert 25 : 1

Local Variable

These variables are declared inside the method of the class. Their scope is limited to the method which means that you can't change their values and access them outside of the method.

In this example, the instance variable has been declared with the same name as the local variable, this is to demonstrate the scope of local variables.

Example of Local Variable



```
1 public class LocalVar {
2     // instance variable
3
4     public String myVar="instance variable";
5
6     public void myMethod(){
7         // local variable
8
9         String myVar = "Inside Method";
10        System.out.println(myVar);
11    }
12
13    public static void main(String args[]){
14        // Creating object
15
16        LocalVar obj = new LocalVar();
17
18        /* We are calling the method, that changes the
19         * value of myVar. We are displaying myVar again after
20         * the method call, to demonstrate that the local
21         * variable scope is limited to the method itself.
22         */
23
24        System.out.println("Calling Method");
25
26        obj.myMethod();
27
28        System.out.println(obj.myVar);
29    }
30 }
31
32
33
34
```

Problems Javadoc Declaration Search Console X Git Staging

<terminated> LocalVar [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/bin/java (Jun 28, 2019, 3:16:01 PM)

Calling Method
Inside Method
instance variable

