Java Variables

Description

- Variable is the name of the kept back region allotted in memory. It can declare based on any Primitive or Reference types prescribed by Java language.
- **Syntax** for Variable declaration is:

```
Data type variable_name [=value];
```

- More than a single variable can be declared for the same data type, in the same line.
- When a variable is declared, the computer allocates memory block, named by that variable, according to the variable's data type.
- This block is stored with the value of that variable.

Types of variables:

1. Local variables
2. Instance variables
3. Static variables

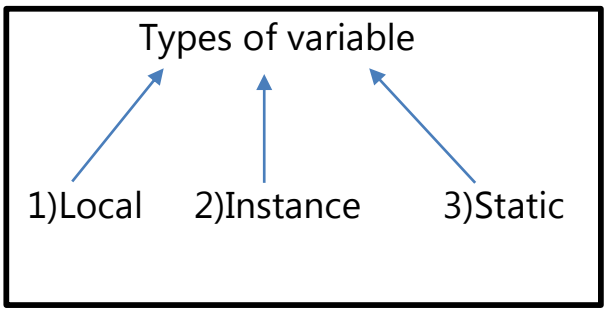


Fig1. Variable Types

Local Variable:

- Declared inside the method / constructor.
- Holds life till the survival of that particular method / constructor.
- Cannot be accessed outside of its method / constructor.
- Must be allotted a value soon after its declaration.
- Memory allocations are executed inside the stack.
- Access modifiers are not applicable due to its limited scope.

Instance Variable:

- Declared outside the method but inside the class.
- Created when an object is formed using 'new' keyword and shattered when the object is destroyed.
- Can be referenced anywhere all over the class.
- Memory allocations are executed in the heap.
- Free to be declared before or after its usage.
- Access modifiers are applicable.
- Has a default value as null.

- Called by name at the place of access in non-static methods.
- Called by fully qualified name like ObjectReferenceName.InstanceVariableName within static methods.

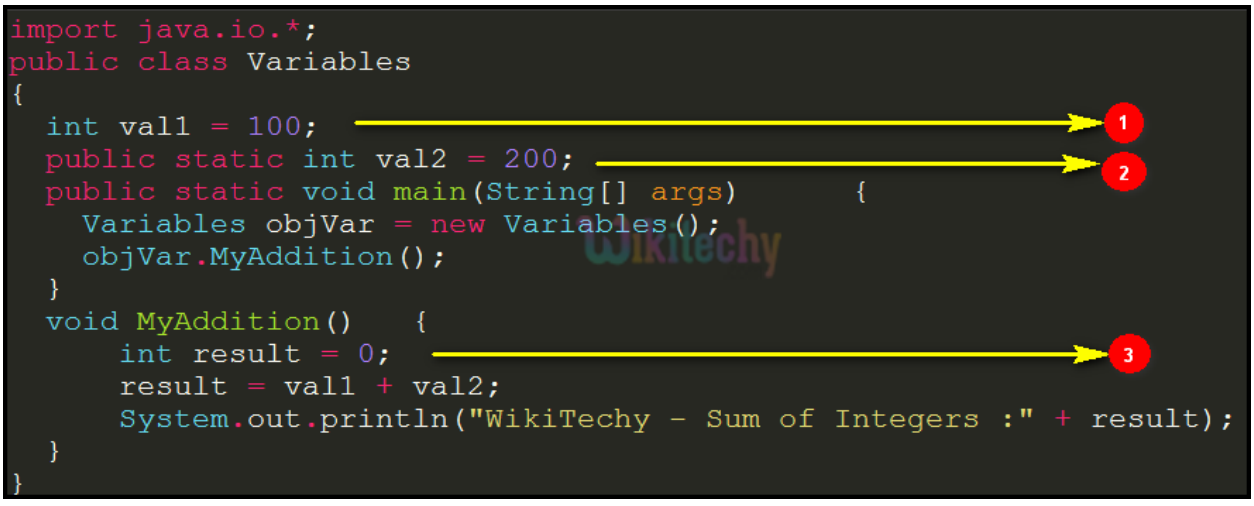
Static variable:

- Declared with the static keyword inside the class, but inside a method.
- Created when the program execution starts and shattered when the program execution ends.
- Seldom used.
- Always declared as constant and hence never have a change in its value.
- Can be referenced anywhere all over the class, similar to instance variable.
- Memory allocations are executed in static memory.
- Access modifiers such as public / private and final are applicable.
- Default values are allotted as per the variable type.
- Accessed by the user using ClassName.VariableName.

Sample Code:

```
import java.io.*;
public class Variables
{
  int val1 = 100;
  public static int val2 = 200;
  public static void main(String[] args)
 {
        Variables objVar = new Variables();
        objVar.MyAddition();
  }
  void MyAddition()
  {
     int result = 0;
     result = val1 + val2;
     System.out.println("WikiTechy - Sum of Integers :" + result);
   }
```

Code Explanation:



1. Instance Variable – has life all over the class "Variables".

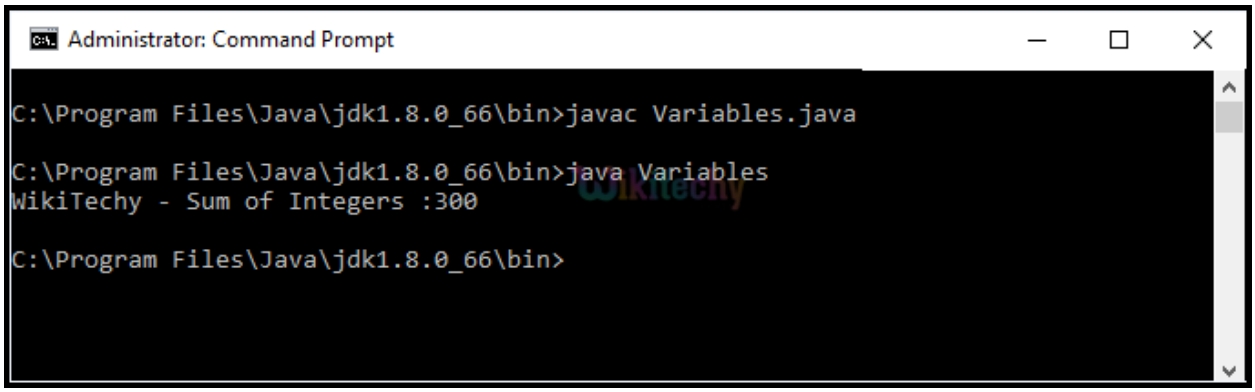
2. Static Variable – has life all over the class "Variables".

3. Local Variable – has life within the method "MyAddition()".

Sample Code:

```
import java.io.*;
public class Variables
{
  int val1 = 100;
  public static int val2 = 200;
  public static void main(String[] args)       {
    Variables objVar = new Variables();
    objVar.MyAddition();
  }
  void MyAddition()    {
      int result = 0;
      result = val1 + val2;
      System.out.println("WikiTechy - Sum of Integers :" + result);
  }
}
```


Output



- Hence the java variables execution has been done successfully.

