

Java Variable Types

Topics : [JAVA](#)

Written on [December 14, 2022](#)

Java provides three types of variables as below.

- Local variables
- Instance variables
- Class/Static variables.

Local Variables

- Local variables are declared in methods, constructors, or blocks.
- Access modifiers cannot be used for local variables.
- Local variables are visible only within the declared method, constructor, or block.
- Local variables doesn't have default value . Local variables should be declared and should assign value before the first use.

Instance Variables

- Instance variables are declared in a class, but outside a method, constructor or any block.
- When a space is allocated for an object in the heap, a slot for each instance variable value is created.
- Instance variables are created when an object is created with the use of the keyword 'new' and destroyed when the object is destroyed.
- Instance variables can be declared in class level before or after use.

- Access modifiers like public, private can be given for instance variables.
- Instance variables have default values. For numbers, the default value is 0, for Booleans it is false, and for object references it is null. Values can be assigned during the declaration or within the constructor.
- The instance variables are visible for all methods, constructors and block in the class. Normally, it is recommended to make these variables private (access level). However, visibility for subclasses can be given for these variables with the use of access modifiers.

Class/Static Variables

- Class variables also known as static variables are declared with the static keyword in a class, but outside a method, constructor or a block.
- Static variables are rarely used other than being declared as constants. Constants are variables that are declared as public/private, final, and static. Constant variables never change from their initial value.
- Static variables are stored in the static memory. It is rare to use static variables other than declared final and used as either public or private constants.
- Static variables are created when the program starts and destroyed when the program stops.

Example :

```
import java.io.*;

/***** Local variables *****/

public class Addition {
public void sum() {
int a = 10;
a = a + 5;
System.out.println("Addition is : " + a);
}

public static void main(String args[]) {
Addition a1 = new Addition();
a1.sum();
}
}

/***** Instance Variables *****/
```

```

public class School {

// this instance variable is visible for any child class.
public String name;

// marks variable is visible in School class only.
private double marks;

// The name variable is assigned in the constructor.
public School (String studName) {
name = studName;
}

// The marks variable is assigned a value.
public void setmarks(double studMark) {
marks = studMark;
}

// This method prints the School details.
public void printStudent() {
System.out.println("Student name : " + name );
System.out.println("Student marks : " + marks);
}

public static void main(String args[]) {
School stud = new School("John roy");
stud.setmarks(70);
stud.printStudent();
}
}

/*****Static Variables *****/
import java.io.*;
public class Fruit {

// fruit_name variable is a private static variable
private static String fruit_name;

// STORE is a constant
public static final String STORE = "ahmedabad ";

public static void main(String args[]) {
fruit_name = "Apple";
System.out.println( "Store name:" + STORE + "Fruit name:" + fruit_name);
}
}

```

Output :

Addition is : 15

Student name : John roy

Student marks :70.0

Store name:ahmedabad Fruit name:Apple

© Copyright **Aryatechno**. All Rights Reserved. Written tutorials and materials by [Aryatechno](#)

ARYATECHNO