01

INTRODUCTION TO OOP AND BASIC JAVA PROGRAMMING.

OBJECT ORIENTED PROGRAMMING LAB

SESSION - 01

CSE215 | DAFFODIL INTERNATIONAL UNIVERSITY

Java Naming Convention:

Java naming convention is a rule to follow as you decide what to name your identifiers such as class, package, variable, constant, method etc.

Name	Convention
class name	**should start with uppercase letter and be a noun e.g. String, Color, Button, System, Thread etc.**
interface name	**should start with uppercase letter and be an adjective e.g. Runnable, Remote, ActionListener etc.**
method name	**should start with lowercase letter and be a verb e.g. actionPerformed(), main(), print(), println() etc.**
variable name	**should start with lowercase letter e.g. firstName, orderNumber etc.**
package name	**should be in lowercase letter e.g. java, lang, sql, util etc.**
constants name	**should be in uppercase letter. e.g. RED, YELLOW, MAX_PRIORITY etc.**

CamelCase in java naming conventions

Java follows camelcase syntax for naming the class, interface, method and variable.

If name is combined with two words, second word will start with uppercase letter always e.g. actionPerformed(), firstName, ActionEvent, ActionListener etc.

Let us now briefly look into what do class, object, methods, and instance variables mean.

- **Object** − Objects have states and behaviors. Example: A dog has states - color, name, breed as well as behavior such as wagging their tail, barking, eating. An object is an instance of a class.

- **Class** − A class can be defined as a template/blueprint that describes the behavior/state that the object of its type supports.

- **Methods** − A method is basically a behavior. A class can contain many methods. It is in methods where the logics are written, data is manipulated and all the actions are executed.

- **Instance Variables** − Each object has its unique set of instance variables. An object's state is created by the values assigned to these instance variables.

• First Java Program

Let us look at a simple code that will print the words ***Hello World***.

Example

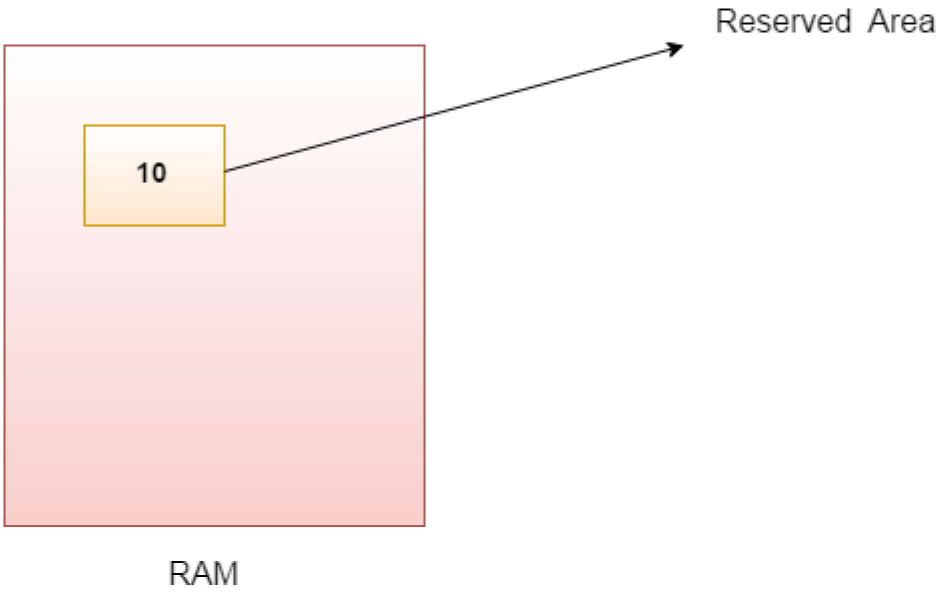
```
public class MyFirstJavaProgram {

   /* This is my first java program.
    * This will print 'Hello World' as the output
    */

   public static void main(String []args) {
      System.out.println("Hello World"); // prints Hello World
   }
}
```

Variable:

Variable is name of *reserved area allocated in memory*. In other words, it is a *name of memory location*. It is a combination of "vary + able" that means its value can be changed.

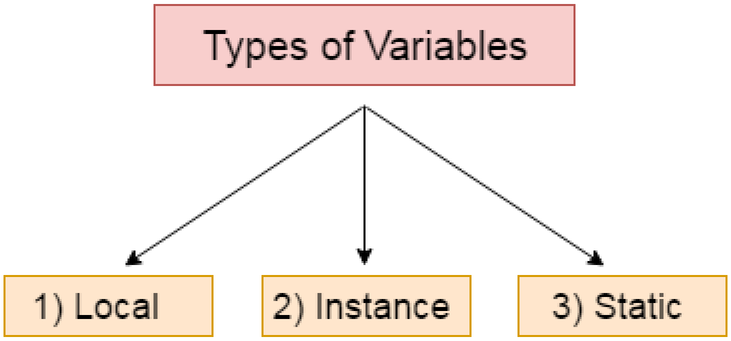


```
int data=50;//Here data is variable
```

Types of Variable

There are three types of variables in java:

- local variable
- instance variable
- static variable



1) Local Variable

A variable which is declared inside the method is called local variable.

2) Instance Variable

A variable which is declared inside the class but outside the method, is called instance variable . It is not declared as static.

3) Static variable

A variable that is declared as static is called static variable. It cannot be local.

We will have detailed learning of these variables in next chapters.

Example to understand the types of variables in java

```java
class A{
int data=50;//instance variable
static int m=100;//static variable
void method(){
int n=90;//local variable
}
}//end of class
```

Simple Java Program: Add Two Numbers

```java
class Simple{
public static void main(String[] args){
int a=10;
int b=10;
int c=a+b;
System.out.println(c);
}}
```

Output:

```
20
```

Java IF-else Statement:

The Java if-else statement also tests the condition. It executes the *if block* if condition is true otherwise *else block* is executed.

Syntax:

```
if(condition){
//code if condition is true
}else{
//code if condition is false
}
```

Java For Loop Example:

```
public class ForExample
{
    public static void main(String[] args)
    {
        for(int i=1; i<=10; i++)
        {
            System.out.println(i);
        }
    }
}
```

- ## Java Array Example:

  - Example on declare, instantiate, initialize and traverse an array.

```
class Testarray{
    public static void main(String args[]){
        int a[]=new int[5]; //declaration and instantiation
        a[0]=10;  //initialization
        a[1]=20;
        a[2]=70;
        a[3]=40;
        a[4]=50;
       //printing array
        for(int i=0; i<a.length; i++) //length is the property of array
            System.out.println(a[i]);
    }
}
```

```
Output:
    10
    20
    70
    40
    50
```

- Example on showing how to create, initialize, and process arrays

```
public class TestArray {

   public static void main(String[] args) {
      double[] myList = {1.9, 2.9, 3.4, 3.5};

      // Print all the array elements
      for (int i = 0; i < myList.length; i++) {
         System.out.println(myList[i] + " ");
      }

      // Summing all elements
      double total = 0;
      for (int i = 0; i < myList.length; i++) {
         total += myList[i];
      }
      System.out.println("Total is " + total);

      // Finding the largest element
      double max = myList[0];
      for (int i = 1; i < myList.length; i++) {
         if (myList[i] > max) max = myList[i];
      }
      System.out.println("Max is " + max);
   }
}
```

This will produce the following result −

Output

```
1.9
2.9
3.4
3.5
Total is 11.7
```

```
Max is 3.5
```