Top of Form

##### **Lecture Description:**

In this lecture, we will discuss:

• Java Basic Syntax

• Various Java keywords

• Identifiers and data types

Bottom of Form

Top of Form

Bottom of Form

Top of Form

#### Java Basic Syntax:

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

**Class -** A class can be defined as a template/ blue print that describe the behaviors/states that object of its type support.

**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.

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

#### First Java Program:

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

**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**

**}**

**}**

About Java programs, it is very important to keep in mind the following points.

**Case Sensitivity -**Java is case sensitive which means identifier **Hello** and **hello** would have different meaning in Java.

**Class Names -**For all class names the first letter should be in Upper Case.

If several words are used to form a name of the class each inner words first letter should be in Upper Case.

Example class MyFirstJavaClass

**Method Names -**All method names should start with a Lower Case letter.

If several words are used to form the name of the method, then each inner word's first letter should be in Upper Case.

Example public void myMethodName()

**Program File Name -**Name of the program file should exactly match the class name.

When saving the file you should save it using the class name (Remember java is case sensitive) and append '.java' to the end of the name. (if the file name and the class name do not match your program will not compile).

Example: Assume 'MyFirstJavaProgram' is the class name. Then the file should be saved as'MyFirstJavaProgram.java'

**public static void main(String args[]) -** Java program processing starts from the main() method which is a mandatory part of every Java program.

#### Java Identifiers:

All Java components require names. Names used for classes, variables and methods are called identifiers.

In Java there are several points to remember about identifiers. They are as follows:

* All identifiers should begin with a letter (A to Z or a to z), currency character ($) or an underscore (\_).
* After the first character identifiers can have any combination of characters.
* A key word cannot be used as an identifier.
* Most importantly identifiers are case sensitive.
* Examples of legal identifiers: age, $salary, \_value, \_\_1\_value
* Examples of illegal identifiers: 123abc, -salary

Bottom of Form

Top of Form

#### ****Java Keywords:****

The following list shows the reserved words in Java. These reserved words may not be used as constant or variable or any other identifier names.

Bottom of Form

Top of Form

##### **Java Data Types:**

There are two data types available in Java:

* Primitive Data Types
* Non-Primitive Data Types

#### ****Primitive Data Types:****

A primitive data type specifies the size and type of variable values, and it has no additional methods. There are eight primitive data types in Java:

#### ****Non-Primitive Data Types****

Non-primitive data types are called **reference types** because they refer to objects.

##### **Difference:**

The main difference between **primitive** and **non-primitive** data types are:

* Primitive types are predefined (already defined) in Java. Non-primitive types are created by the programmer and are not defined by Java (except for String).
* Non-primitive types can be used to call methods to perform certain operations, while primitive types cannot.
* A primitive type has always a value, while non-primitive types can be null.
* A primitive type starts with a lowercase letter, while non-primitive types start with an uppercase letter.
* The size of a primitive type depends on the data type, while non-primitive types have all the same size.

Examples of non-primitive types are Strings, Arrays, Classes, Interface, etc.

Bottom of Form

Top of Form

Bottom of Form

Top of Form

##### **Reference Links:**

**ONLINE NOTES LINKS:**

<https://www.w3schools.com/java/java_ref_keywords.asp>

<https://www.geeksforgeeks.org/list-of-all-java-keywords/>

<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/_keywords.html>

<https://www.w3schools.com/java/java_data_types.asp>

<https://www.geeksforgeeks.org/data-types-in-java/>

<https://www.youtube.com/watch?v=bqPIWlnjWbA>

<https://www.youtube.com/watch?v=iFzA43xR04s>

**VIDEO LINKS:**

<https://marcus-biel.com/basic-java-keywords-explained-debriefing/>

<https://www.youtube.com/watch?v=f8lCiYquj28>

<https://www.youtube.com/watch?v=e5ftceLyKbM>

<https://www.youtube.com/watch?v=TL7tdNp0raE>

<https://www.youtube.com/watch?v=Jiy5nbDjtFM>

Bottom of Form

Help for current page