Full Stack Development

===============================

**Front End**

**Back End (Java)**

Core Java language

- Basics of Core Java

Intro to Java, Installation and env setup, Writing, Compiling and executing first java program, Data Type, Control Flow statements, Array, class, Object, method, String classes, User inputs.

- OOPs in Java

**Encapsulation, Inheritance, Polymorphism, Abstraction**, constructor, Keyword in Java, Package, imports, access modifiers.

- Advance Concepts in Core Java

Exception Handling, Collection, threading, JDBC

Advance Java

- Servlet technologies

- JSP technologies

Spring Boot Framework

- REST API

**Database (MySql)**

- SQL (Structure Query Language)

**DevOps Tools**

- Maven

- Git

- GitHub

**Software**

JDK

Mysql Database

Eclipse IDE

Git

Maven

Postman

**Java Setup**

1. Download JDK

<https://www.oracle.com/java/technologies/downloads/>

Download the Installer for you operating System.

1. Install JDK
2. Verify the Installation
   1. You can verify the installation file into C:\Program File\java\<jdk-version-folder>



1. Setup Environment Variable
   1. **Set JAVA\_HOME**
      1. Go to Start and Search for “Environment” word and select the “Edit System Environment Variable” option.
      2. Click on the “Environment Variable” button on the new window.
      3. Select “New” Button from the ‘System Variables’ section
      4. Provide the following details into text box
         1. Variable Name: **JAVA\_HOME**
         2. **Variable Value: <JDK-Path>**



* 1. **Set Path**
     1. Check for “path” variable inside ‘system variables’ section
     2. Select “Path” variable and click on “Edit” Button
     3. Click the “New” button on the new window
     4. And set the variable value as follows

**%JAVA\_HOME%\bin**

* + 1. Click on “OK”



1. Verify the Environment Variable Setup
   1. Open CMD
   2. Try the following commands
      1. **java -version**
      2. **javac**



Java Introduction.

1. It is introduced in 1990. Initially it is known as Oak Language. Then it is renamed as Java language.
2. From Java 2nd version onwards, java language distributed into 3 editions.
   1. Java SE (J2SE)
      1. Java SE stands for Java Standard Edition.
      2. All the basics of java concepts included in this edition and it is also known as Core Java.
      3. Can Develop Console Based and Desktop Application.
   2. Java EE (J2EE)
      1. Java EE stands for Java Enterprise Edition.
      2. It is a combination of multiple java technologies. It is also known as Advance Java.
      3. It is majorly used for developing a Web Application.
   3. Java ME (J2ME)
      1. Java ME stands for Java Micro Edition.
      2. Using this you can develop Embedded and Mobile application.
3. Initially java language was managed by Sun Microsystem which acquire by Oracle on 20s.

**Important Components of Java**

1. JVM
   1. JVM stands for Java Virtual Machine.
   2. It is use to execute the java program, and also support the all activities required for execution such as stating up of you code, memory allocation, communicating with OS.
   3. To Execute the java program JVM must be present inside you system.
2. JRE
   1. JRE stands for Java Runtime Environment.
   2. It is consisting of JVM and APIs. Which create the runtime environment for the java program.
   3. On users system JRE must be available.
3. APIs
   1. APIs stands for Application Programming Interface.
   2. It is a set of predefine functionalities which is provided by java language.
4. JDK
   1. JDK stands for Java Development Kit.
   2. It is a combination of JRE, JVM, APIs and some development tools.
   3. You can develop the java code, compile it and execute the java code.
   4. JDK has present in Developers system.

**Writing and Executing First Java Program**

1. Writing a java code
   1. You can write a using IDE or Notepad or any editor tool.
2. Java Program Syntax
   1. Create Java class
      1. In java everything must be write inside class except import and package statement.
      2. Syntax:

**public class <Class\_Name>**

**{**

**}**

* 1. Cerate Main Method inside java class
     1. Main method is the first method which gets executed.
     2. This method is a start point of java program.
     3. This method will be called internally by java (JVM) when you execute your program.
     4. Syntax:

**public static void main(String args[])**

**{**

**}**

* 1. Write an executable code into method.

**System.out.println(“Hello, Welcome to first java program”);**

1. Save File
   1. File Name must be same as public class name.
   2. File extension must be **.java**.
2. Compile the java code
   1. In this sept the java code will be converted into a byte code and the binary file will be created.
   2. As an outcome of the compilation step the .class file will be created.
   3. In this step all the syntax validation will also happens.
   4. To compile a code first Open a CMD
      1. The path of the CMD must be pointing to a location where you save your java class. (to do this you can go to a folder where you save your java class, click on the address bar and type “cmd” and hit enter)
      2. Execute a command to compile

**javac FileName.java**

1. Execute Java code
   1. Use same CMD
   2. Execute a command to run java program

**java className**

Rules:

1. In a source file you can create more than one classes. But only one class must be public and the file name must be same as public class name.
2. In all the classes created in a source file can have a separate main method.

**Keyword, Literal and Identifier**

**Keyword:**

1. Keywords are the reserve words by the java programming language.
2. There are about 52 keyword present in java.
3. Every keyword is in small case.
4. Example:

public, class, void, static, protected, private, default, if, else, switch, where, for, do, while, int, short, long, byte, double, float, boolean, char, null, true, false etc.

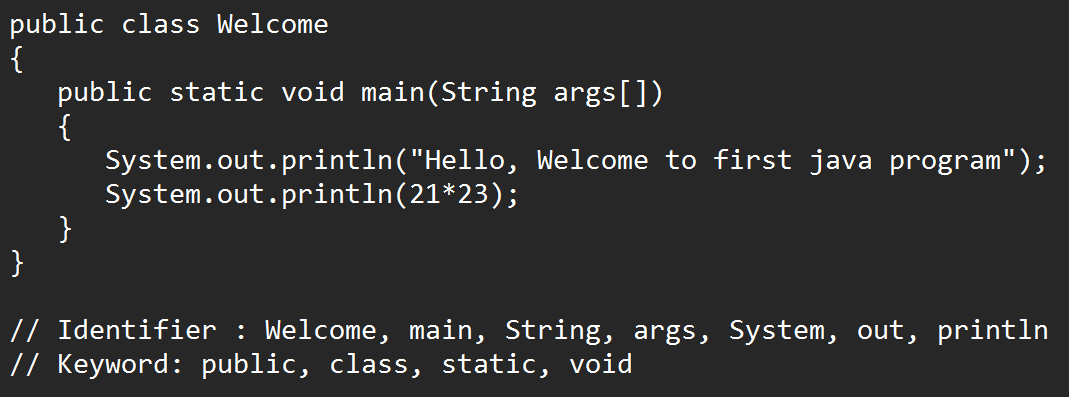
**literal:**

1. Literal are also consider as a values. Some values are fixed by java which is also in the list of keyword.
2. String values are also known as literals.
3. Example:

null, true, false

**Identifier:**

1. Identifier are the words which is used by developer for identifying the program components.
2. The identifiers are used to create variable, method, class, object names.
3. Example:



1. Rules to create identifier:
   1. Identifier must not be a keyword.
   2. Identifier can contains alphabets, Number and symbols.
   3. Only **$** and **\_** symbols are allowed in identifier. No other symbol is allowed as an identifier.
   4. Identifier must be start with alphabet or symbols. Identifier must not be start with number.
   5. Identifiers are case sensitive and must not contains spaces.
   6. There is no character limit to create identifier.
2. Conventions to create method, variable and object name.
   1. The name should be start with small case if it is a combination of multiple words then every word start with capital case from the 2nd word onwards.
   2. Example: employeeId, studentName, printEmployeeDetails
3. Convention to create class.
   1. The class name should be starts with capital case and if it is a combination of multiple words then every word should be start with capital.
   2. Example: Welcome, EmployeeDetails, AdminAddressDetails etc.
4. Convention to create Constants:
   1. Constants must be in capital case.
   2. Example: GRAVITY, PI etc.

**Data Type and Variable**

**DataType:**

1. Data Types are required to specify the type of values used in a program.
2. Data types are used to create a variable.
3. There are 2 types of data type in java
   1. Primitive Data Type
      1. Primitive Data type has a fixed sized which is predefine in java.
      2. Primitive data type are in non-object format.
   2. Non-Primitive Data Type
      1. Non-primitive data type doesn’t have any fixed size, its size will be decided as per it uses in the code.
      2. Non primitive data types are always in the Object format.

