JAVA

Why JAVA?

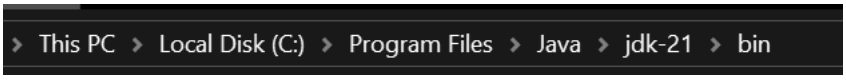
1. Platform Independent or architecture independent (WORA)
   Platform -> Combination of Operating System + underlined hardware

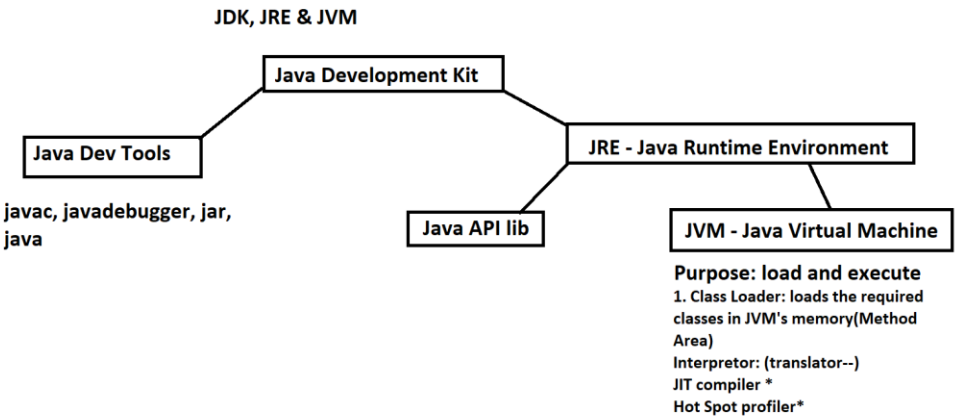
   C&C++
   C compiler produces an executable file which run on only specific platform.
   If we compile C code on windows it can run on only windows.

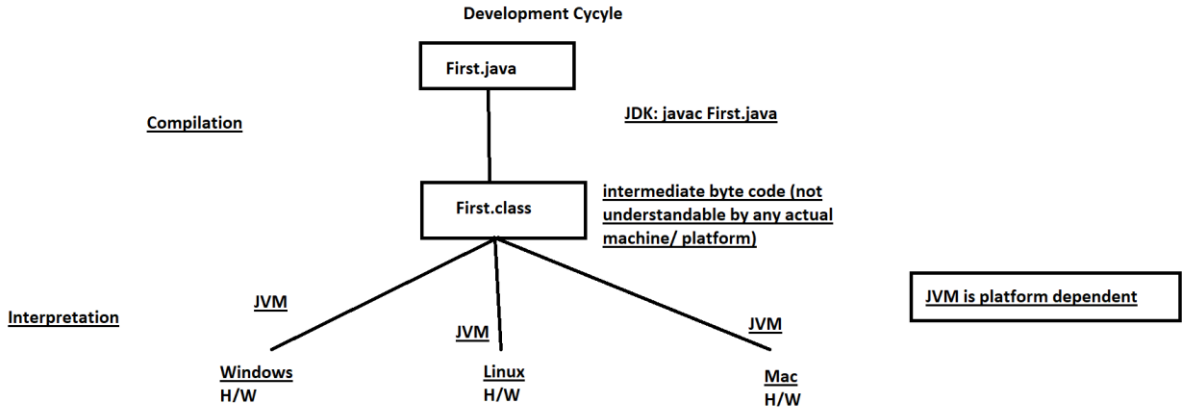
   - Extension of source file for java -> .java
   - Extension of compiler -> .class
     (intermediate byte code)

   WORA-> Write Once and Run Anywhere
   Run on any database
   Run on any web server

   Installed JDK and set the environment variables







2.

3. Simple & Robust (Strong & Failproof)

   Robust:
   - Automatic Memory Management
   - Exception Handling
   - Platform Independent
   - Thread Management

4. Secure
5. Automatic Garbage Collection
   In C++ if we are creating an object but no longer need it, we need to clean it from the memory (By destructor or delete)
   Java -> Garbage Collector
6. Multi-threaded Support
7. Object Oriented Support
   - Why java is not completely object oriented?
     ➔ For the purpose of efficiency, java supports primitive data types.
     ➔ Primitive data types are not object types.

DOWNLOAD AND INSTALL IDE

- Structure of JAVA program
  1. Main Method:
     Entry point of java program.
     public static void main(String[] args)
     {//Code to be executed}
     public: Access Modifier that makes the method accessible from anywhere
     static: Allows the method to be called without creating the object.
     void: Indicates that the method does not return any value.
     String[] args: Parameter to receive command-line arguments

- Variables:
  What?
- A container which holds the data that can change during the execution.
  Syntax:
  dataType variableName = value;

- DataTypes
  Java has 8 primitive datatypes
  1. int: used for integers
  2. float: Used for decimal numbers
  3. char: Used for single character
  4. boolean: used for true/ false values
  5. byte: used for small integers
  6. short: used for small integers
  7. long: Used for large integers
  8. double: Used for large decimal numbers

Declaring and Initializing variables

- Declaration: int x;
- Initialization: x = 10;
- Combined: int x = 10;

```
package examplesFeb;

public class Main {
    public static void main(String[] args) {
        int age = 25; // Integer
        float price = 99.9f; //Float (f suffix is
mandatory)
        char grade = 'A'; //Character
        boolean isEligible = true; // Boolean

        System.out.println("Age = "+ age);
        System.out.println("Price = "+ price);
        System.out.println("Grade = "+ grade);
        System.out.println("Are you eligible? "+
isEligible);

    }
}
```

Comments:

1. Single line comment: Starts with //
2. Multi-Line comment: Starts with /*
   ends with */

Operators:
What?
Are nothing but symbols which performs operations on variables or values.

Categories:

1. Arithmetic Operators (+,-,*, /, %)
2. Relational Operators (==, !=, >,<,<=,>=)
3. Logical Operator(&&, ||,!)