What is Java?

* It is a programming language.
* It is high level, robust, object oriented and secure programming language.
* It is platform independent. It was introduced in 1995.

<https://docs.oracle.com/javase/8/docs/api/index.html>

OpenJDK

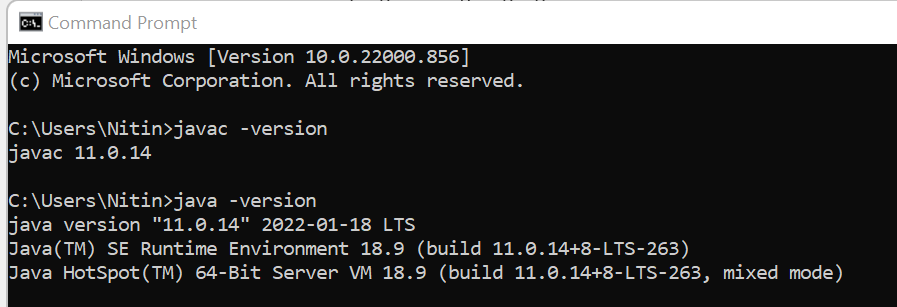
<https://www.oracle.com/java/technologies/downloads/>

Oracle JDK

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

after installation - verify the installation

go to command prompt / terminal and ensure the version for javac & java are the same.



If it doesn’t work,

Set the environment variable that is PATH.

set PATH = %PATH%; C:\Program Files\Java\jdk-11.0.14\bin

alternatively we can go to Edit System Variable and add Java Home path to Path variable.

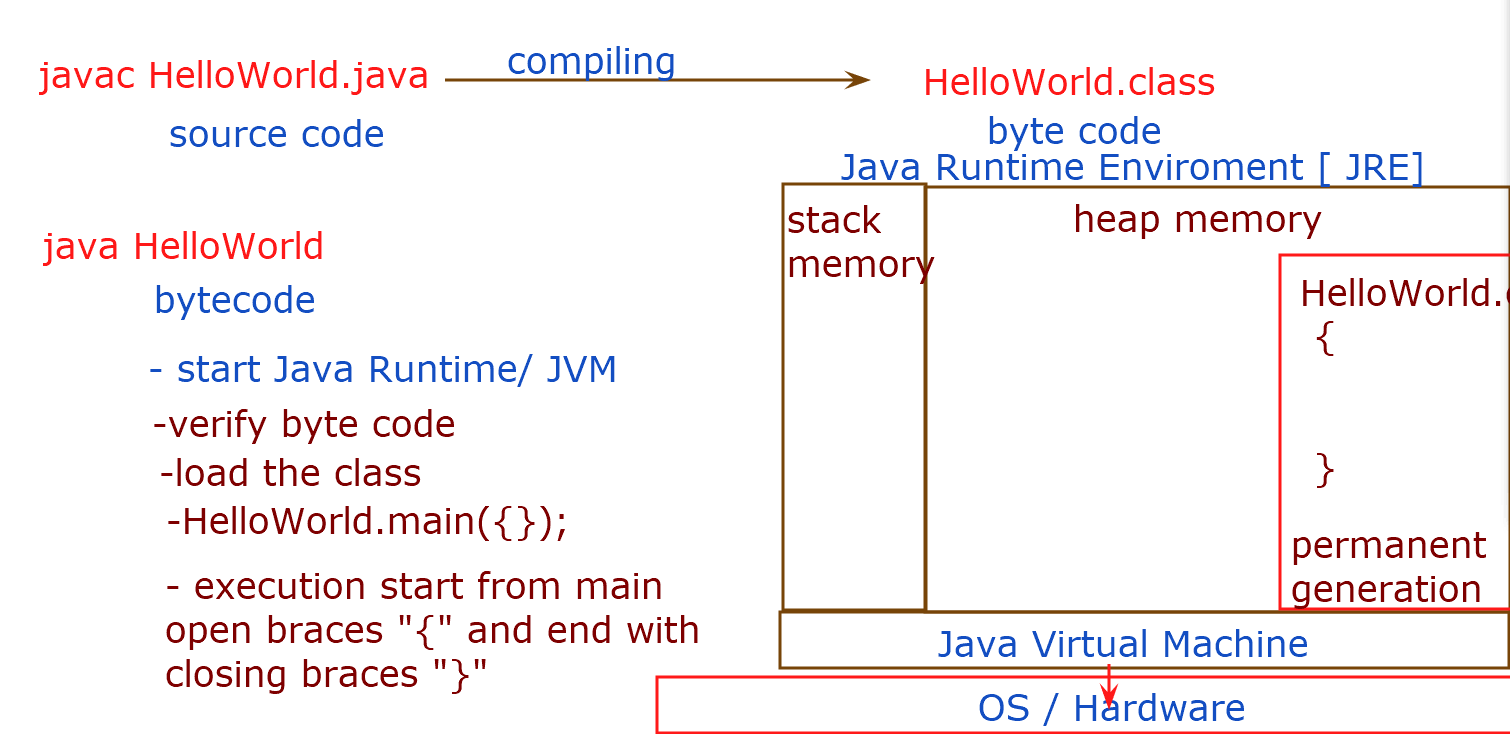
MacOS [ pls check the java home path ]

export PATH = $PATH: /usr/java/jdk-11.0.14/bin

javac === Java compiler / it is used to translate high-level Language to Byte Code.

java == Java Runtime / it is responsible for starting JVM and executing Java Main class that is the starting point of your java application.

**Write Once and Run Anywhere**



**Types of Java Edition**

1. Java Standard Edition [ Java SE ]
   * Desktop applications
   * Console Application [ No GUI ]
   * Java Libraries
2. Java Enterprises Edition [ Java EE ]
   * Web Application
   * Distributed Applications [ Enterprise Application]
3. Java Micro Edition [ Java ME ]
   * Mobile Application
   * Java TV
   * Java Card [ program information for smart card chipset]

**Object Oriented Programming**

1. Encapsulation & Data Hiding
2. Inheritance
3. Abstraction
4. Polymorphism
5. **Encapsulation & Data Hiding**

Wrapping up data & behavior into one capsule / block is known as encapsulation. A class is a way to encapsulate the state[ data ] & behavior.

This is to form blue print.

Data hiding - To hide the data from direct accessibility to prevent direct modification.

Graphical user interface, text

Description automatically generated

**Syntax for Creating Object**

<ClassName> ref-variable = new <ClassName>();

* Java is case Sensitive
* Java follows certain naming convention

1. ClassName - upper camel case
2. Field members [ state ] & methods [ behavior ] – lower camel case
3. Constant variable -- UPPER CASE

Text

Description automatically generated

**Types of Variables**

1. Primitive Type [ 8 types ]
   1. byte 8 bits -128 to 127
   2. short 16 bits -32768 to 32767
   3. **int**  32 bits -2147483648 to -2147483647
   4. long 64 bits
   5. float 32 bits
   6. **double**  64 bits
   7. char 16 bits Unicode char ‘A’
   8. boolean \*1 bit [ actual size may vary depending on OS] true/false

Any number with a precision literal value like 1.0 , is by default double type

All whole numbers 10 are by default-int

1. Reference Type or Non-Primitive

Class, Interface, Enum & Array types of variables are known as reference types.

Diagram, schematic

Description automatically generated

**Scope of Variables**

1. Instance Variable / Field members – Accessible in all methods in the given class. All the instance variables get initialized with default values at the time of instantiation.

byte, short, int , long ------------------------------ 0 [ default value ]

float, double ------------------------------------------ 0.0

boolean ----------------------------------------------- false

char --------------------------------------------------- ‘\u’

all reference type , the default value is null.

1. Local Variables - Accessible only in the block that they are declared in. you must initialize it before you use it.
2. Static Variables / Class Variable – Accessible by all instances and maintains only copy per JVM.

**Static Variables** – they get memory allocated at the time of class loading. They belong to class memory. It is initialized with a default value.

There is the only copy of a static variable is created per JVM. This can be shared by all instances of a class.

**Static Methods:**

You can use a static keyword to declare a method. This method can access only static variables. You will have only one copy of static context per JVM.

Relational Operators

>

<

<=

>=

==

!=

Arithmetic Operators

+

-

\*

/

%

++

--

Logical Operator

&&

||

!

Access Modifier “private” keyword can be used to hide.

Constructor is a method with the same name as classname and no return type. It is invoked automatically at the time of instantiation. It is used to initialize the fields.

A). No Argument Constructor [ default constructor ] is supplied by the compiler to the byte code of a class only if no constructors were added by the developer.

B) No default constructor will be added if you have your own constructor. You may need to add it manually in your code.

**Inheritance**

**public** **class** Manager **extends** Employee {

**private** String deptName;

**public** Manager(**int** id, String name, **double** salary, String deptName) {

**super**(id,name,salary);

**this**.deptName=deptName;

}

**public** String getDeptName() {

**return** deptName;

}

**public** **void** setDeptName(String deptName) {

**this**.deptName = deptName;

}

}

**Method Overriding**

Re-writing the super class method that is inherited, is known as method overriding.

* Method in subclass must have the same name, signature [ arguments types, numbers and sequence] and return type.
* Overridden method can not have less accessibility than super class method.