**CORE JAVA**

**History:**

* Made in 1995 by Sun Microsystems and lead by the team James Gosling and later by Oracle.
* Java updates in every six months not major but want to make Java as most powerful language.
* It’s an Object Oriented Programming.

**IDE:**

There are many like NetBeans, IntelliJ, Eclipse. Here VS Code is used i.e., Light Weight.

**Compiler:**

**JDK –** Java Development Kit

**Note:** Always download LTS (Long Term Support) despite of new releases.

Open CMD Prompt:

For version, **java --version** and

For Compiler, **javac --version**

If these commands throws as ‘variable not found’ then set path in environmental variables.

Programmer 🡪 Java Code 🡪 Compiler 🡪 ByteCode 🡪 JRE 🡪 JVM (OS+ H/W)

(.java) (javac) (.class)

**Note:**  JVM is dependent on OS and Hardware.

**Variable:**

To store the data.

* Java is also known as Strongly typed language.

**Data Types:**

**Types:**

1. Primitive
2. Non Primitive

**Primitive:**

* Integer
* Float
* Character
* Boolean

**Integer:**

* Byte – 1byte
* Short – 2bytes
* Int – 4bytes
* Long – 8bytes

**Float:**

* Double – 8bytes
* Float – 4bytes

**Note:** By default, double will be used by JavaC.

**Character:**

* Char – 2bytes

Uses UNICODE not ASCII.

Will be in single quotes.

**Boolean:**

* Boolean- True or False

Used for conditions.

**Literal:**

Nothing but a value assigned to a variable.

**Type Conversion and Casting:**

Conversion of one data type to another. a=b 🡪 Implicit

Converting integer value into byte is known as Casting b = (byte)a 🡪 Explicit

byte b = 127;

int a = 12;

**Operators:**

* Assignment operators (=)
* Relational Operators (<, >, ==,!=, <=, >=)
* Logical Operators (&&, ||, !)

**Conditional Statements:**

* If
* If else
* If else if
* Switch

**Ternary Operator:**

?: - Like if else in some cases.

**Loops:**

* While
* Do While
* For

**Objects And Classes:**

* Properties and behaviour
* When we need to create object,first we need to create class.
* Class acts as a blueprint.
* JVM creates objects in Java.

**JDK, JVM And JRE:**

**JDK:**

* Java Development Kit.
* Compiles the code.
* JDK acts like a upper layer.

**JVM:**

* Java Virtual Machine.
* Runs the code.

**JRE:**

* Java Runtime Environment.
* To run the code, we need some extra files like inbuilt classes, JRE provides those extra files.
* JVM is a part of JRE.

JDK

JRE

JVM

**Method Overloading:**

**Method:** block of code runs when it is called

Same method used but with different parameters.

main( ) – is a method and start of the execution.

**Stack:**

Last in First Out

Memory Block

Stores data

Local variables are part of stack.

**Heap:**

Here we have open space to store data.

Memory Block.

Stores data.

Instance variables are part of Heap.

**Local Variables -** Created inside method.

**Instance Variables -** Created inside class.

**Array:**

Holds multiple values.

**String:**

Collection of characters.

**Mutable String** - Changed

**Immutable String** - Unchanged

**String Buffer**

Capacity(Buffer size) – 16 characters . Used when if there is no continuous location of data it relocates in heap.Reduces relocation if there is no space.

**String Builder –**

**Note:** Stringbuffer is thread safe where as String builder is not.

**Static Variable:**

Static variable should be called with calss names but we can call with object names also try to avoid it.

Static are nothing but constant variable through out the program when we assign a value to it.

These are shared by different objects.

Static keyword – something we are making as a class member not an object one.

**Static Method:**

Whenever non static method is used create object of it.

Static method can be called by a class name.

**Encapsulation:**

Binding of data with methods, no one from the outside world can use the variables directly. It is accessible through methods only.

We are encapsulating data and methods – It is known as Encapsulation.