**Java**

* Invented in 1995 by James Groslin. Named after Java Coffee.
* It’s a platform independent programming language.
* Latest edition is Java SE.
* It has byte code class which is platform independent.
* Mechanism of Java

1. Compilation
2. Execution

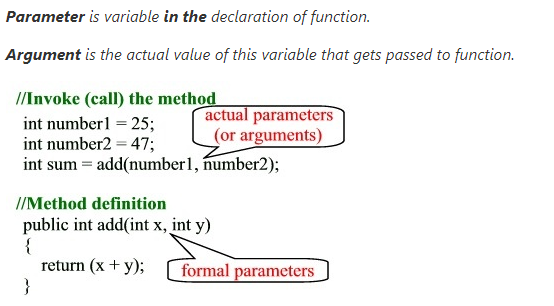
Compilation: Java Raw Code(.java) 🡪 Java Compiler 🡪 Java byte Code(.class)

* Java byte code is recognized by every operating system if it has Java Runtime Environment (JRE)/Java Virtual Machine (JVM).
* We can run our java code from terminal | go to “src” folder and open terminal and type:

>>>java com.packagename.Classname

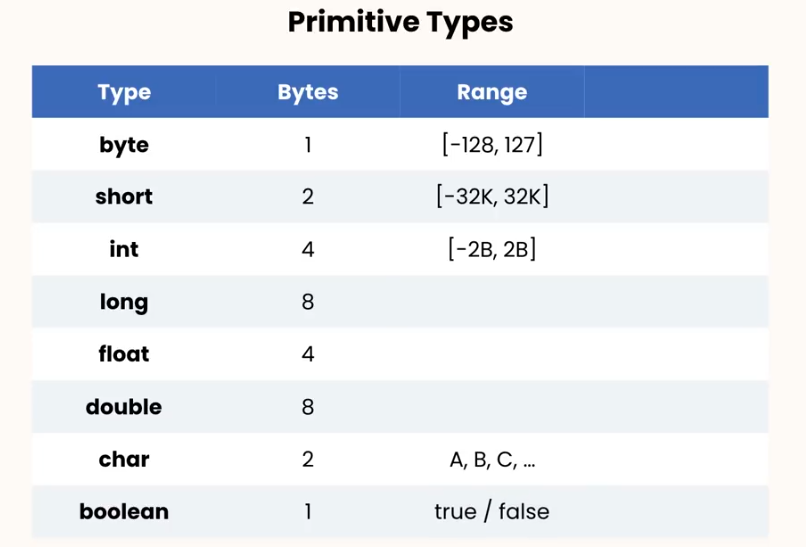
* Method’s curly braces start on the first line.
* Every java program resides into a package.
* Inside every package there must be one class which will be having one must method called main.
* Both method and class start with an access modifier.
* Java’s Phases:

1. Edit
2. Compile
3. Load
4. Verify
5. Execute

* Key point
* Project name, Class name use PNC (Pascal Naming Convention e.g.: CodeWithMosh)
* Method name, Variable name use CNC (Camel Naming Convention e.g.: codeWithMosh)
* Everything that has double first braces as suffix is called method. (e.g.: println ())
* Memories allocated and released by JRE.
* 
* In java strings are immutable that means you can’t change the string value.
* Sout+tab = System.out.println();
* To make a variable constant we need to add ‘final’ keyword as prefix.
* We can use format specifier as like C language.
* Fundamentals:

Types

1. Primitive(variables)

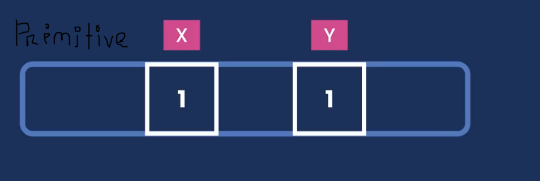


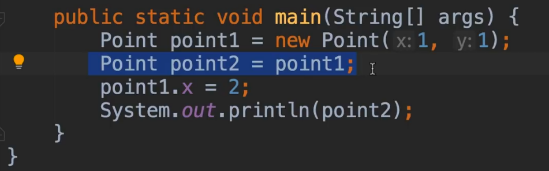
1. Non-primitive/Reference(arrays)

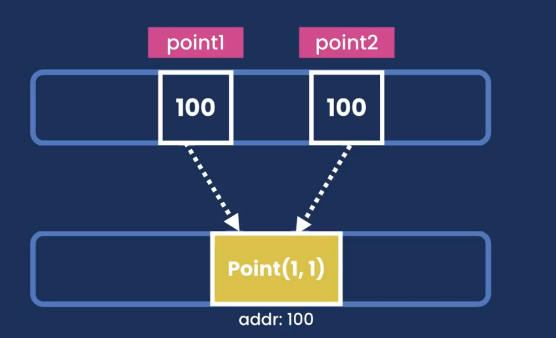
* For reference type we need to use an operator to allocate memory.

Difference between primitive and non-primitive types:

When we declare primitive variables, they store the value but reference type stores address of a point. An example has given below:







* String
* Strings are immutable which means you can’t change a string.
* String.toLowerCase() 🡪 change string character into lowercase.
* String.toUpperCase() 🡪 change string character into uppercase.
* String.trim() 🡪 removes unnecessary spaces from front and end.
* String.replace(target:’x’, replacement:’y’)
* String.startsWith(), String.endsWith() 🡪 returns a Boolean value, True/False.
* String.length() 🡪 find out length.
* String.indexof(‘x’) 🡪 find out the index no of x. If it cannot find out value into the string it returns negative one (-1).
* To use double/single quote, we need to use ( \” or \’ ) before double/single quote.
* To use slash, we need to use double slash (// or \\)
* Arrays
* To print one dimensional array, we need to use “Arrays.toString(name\_of\_array)”.
* To print multi-dimensional array, we need to use “Arrays.deepToString(name\_of\_array)”.
* Casting
* Implicit casting

Short x = 1;

Int y = x + 2;

So, what happens here in the second line, in memory an anonymous variable has been created store short value as integer and then the addition happens.

Byte>short>int>long>float>double

* To convert type, we just need to type the name of that variable before the value in first braces like shown below:

Explicit casting:

double x = 10;

int y = (int) x + 2;

String to int/double:

String x = “1”;

Integer.parseInt(x);

Variables:

* We use variable to temporarily store data.
* Use CNC.
* We can put int value like this “int uncommonValue = 1\_234\_567;”
* If we use “long” data type we have to add “L/l” as suffix (e.g.: long x = 2\_345\_567\_543L;).
* If we use “float/double” data type, we have to add “F/f” as suffix (e.g.: float = 3.45f;).