VTC Academy
Think Ahead, Beyond Boundaries

Introduction to Programming

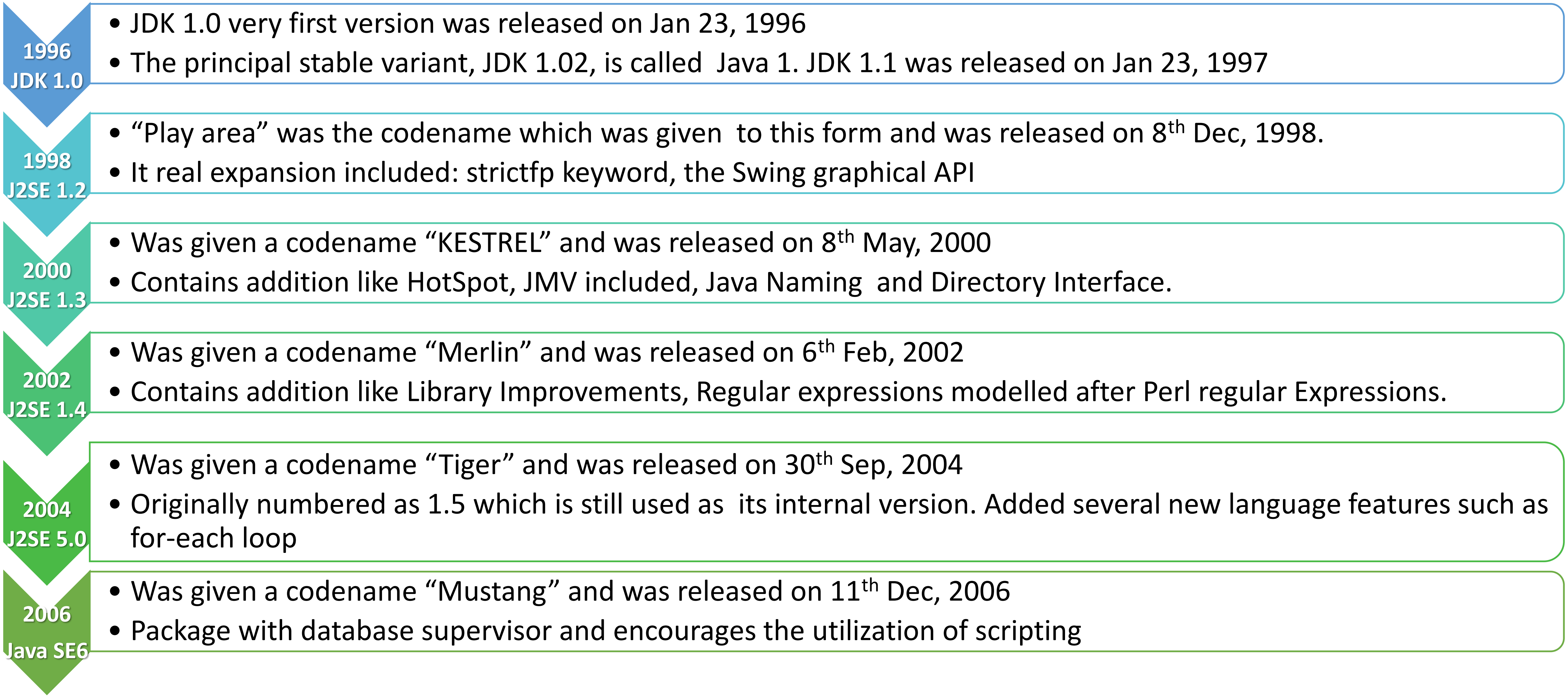
Session 01

Introduction to Programming language

Objectives

- Introduction to Java programming language
- Why Java?
- Java IDE
- Hello, world.
- Basic syntax
- Variable, Constant
- Data types
- Console Input-Output

Java programming language



1996
JDK 1.0

- JDK 1.0 very first version was released on Jan 23, 1996
- The principal stable variant, JDK 1.02, is called Java 1. JDK 1.1 was released on Jan 23, 1997

1998
J2SE 1.2

- “Play area” was the codename which was given to this form and was released on 8th Dec, 1998.
- It real expansion included: strictfp keyword, the Swing graphical API

2000
J2SE 1.3

- Was given a codename “KESTREL” and was released on 8th May, 2000
- Contains addition like HotSpot, JMV included, Java Naming and Directory Interface.

2002
J2SE 1.4

- Was given a codename “Merlin” and was released on 6th Feb, 2002
- Contains addition like Library Improvements, Regular expressions modelled after Perl regular Expressions.

2004
J2SE 5.0

- Was given a codename “Tiger” and was released on 30th Sep, 2004
- Originally numbered as 1.5 which is still used as its internal version. Added several new language features such as for-each loop

2006
Java SE6

- Was given a codename “Mustang” and was released on 11th Dec, 2006
- Package with database supervisor and encourages the utilization of scripting

Java programming language

2011 Java SE7

- Was given a codename “Dophin” and was released on 7th July, 2011
- Add small language changes including string in switch. The JVM was extended with support for dynamic languages

2014 Java SE8

- Was released on 18th Mar, 2014.
- Language level support for lambda expressions, default methods and new date and time API inspried by Joda Time

2017 Java SE9

- Was released on 21th Sep, 2017.
- Projects Jigsaw: designing and implementing a standard, module system for the Java SE platform, and to apply that systems to the platform itself and the JDK

2018 Java SE 10

- Was released on 20th Mar, 2018
- Contains addition like additional Unicode language-tag extenstions , Root certificates, Thread-local handshakes, Heap allocation on alternative memory devices.

2018 Java SE 11

- Was released on 25th Sep, 2018
- Contains addition like Dynamic class-file constant, Epsilon: a no-op garbage collector, local-variable syntax for lambda parameters, Low-overheap heap profiling.

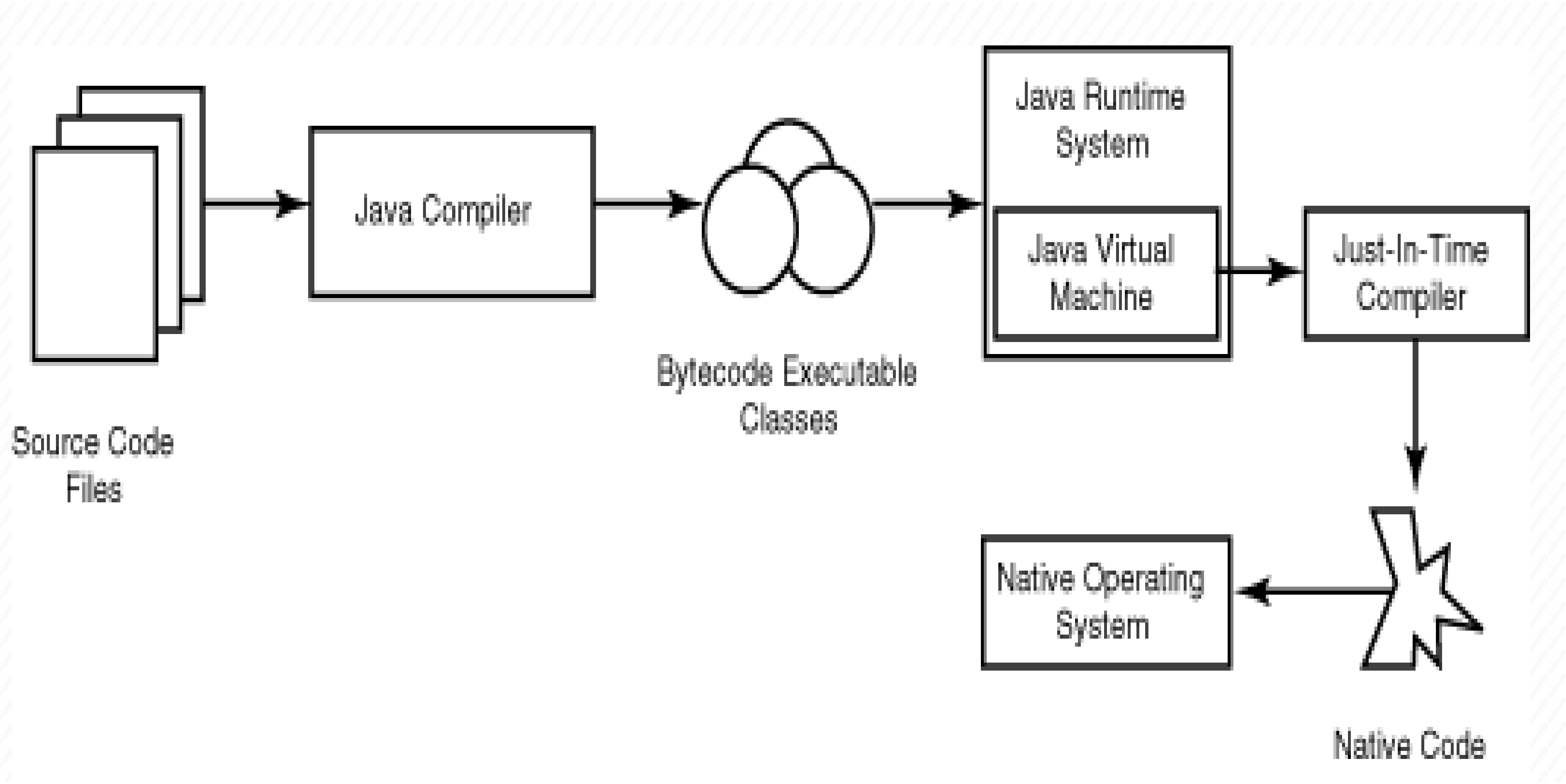
2019 Java SE 12

- Was released on 19th Mar, 2019
- Contains addition like Shenandoah: a Low-Pause-Time Garbage Collector (Experimental) Microbenchmark Suite, Switch Expressions (Preview), JVM constants API.

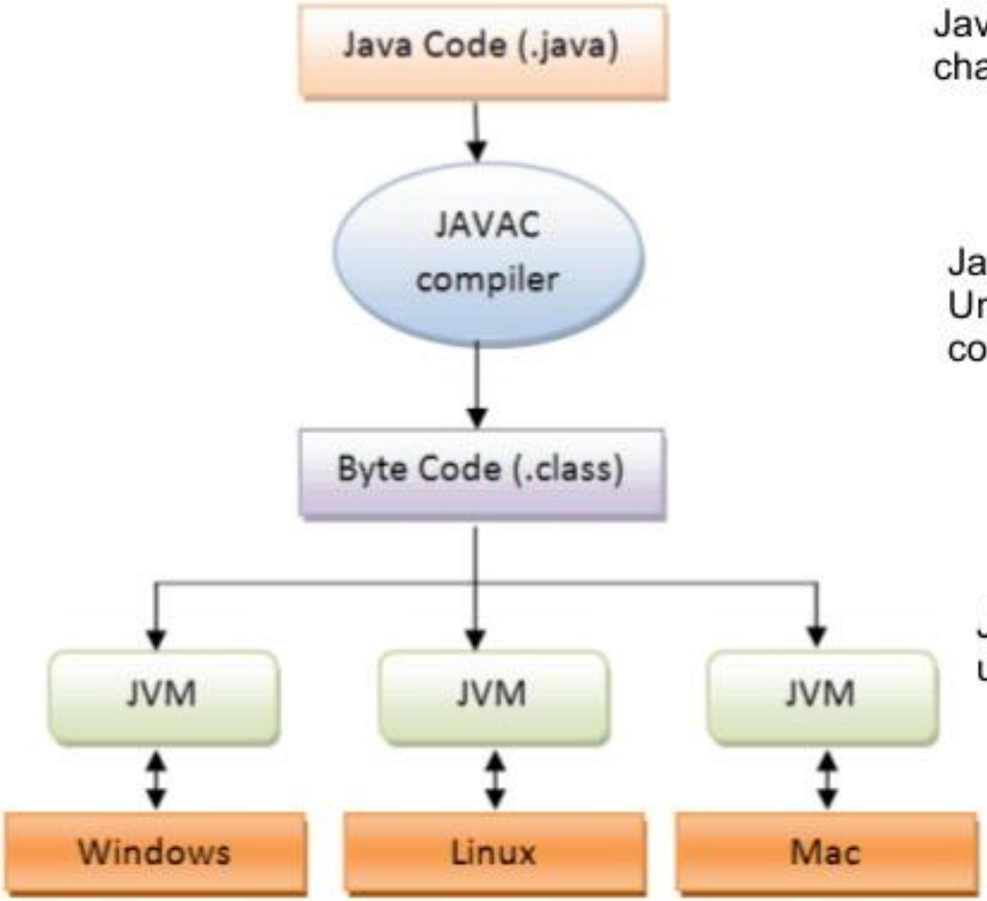
Why Java?

- Java is an object-oriented programming language (OOP), multi-purpose and platform independent. Instead of compiling to machine code on a specific platform, java code is compiled into bytecode – an intermediate format. The bytecode will then be run by the execution environment
- In other words, java code "writes once, runs anywhere". It will work on any device running the operating system, as long as the Java Runtime Environment (JRE) is installed.
- Java is the popular choice for many reasons:
  - employment opportunities
  - free
  - extensive source code gallery
  - support documenting with javadoc
  - strong, dynamic and engaging user community

Compile and Excute program



Compile and Excute program



Java class is written in Unicode characters.

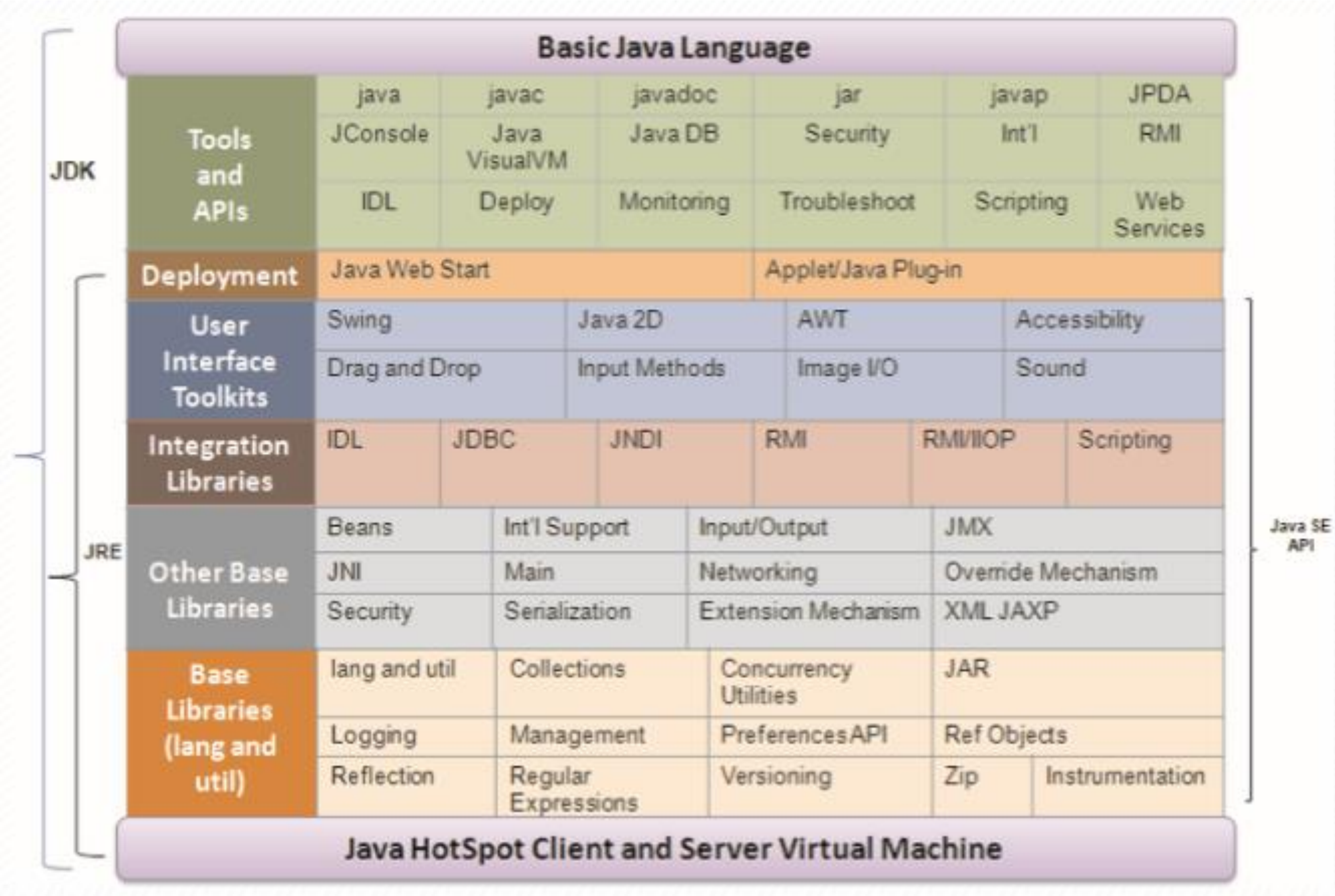
Java compiler convert these Unicode characters into Byte code.

Java Byte code can only be understandable by JVM.

JVM is native code and specific to OS

Java SE Platform

- JRE (Java Runtime Environment): provides JVM (Java Virtual Machine) and library used to run Java programs.
- JDK (Java Development Kit): includes the compiler and debugger used to develop Java applications.

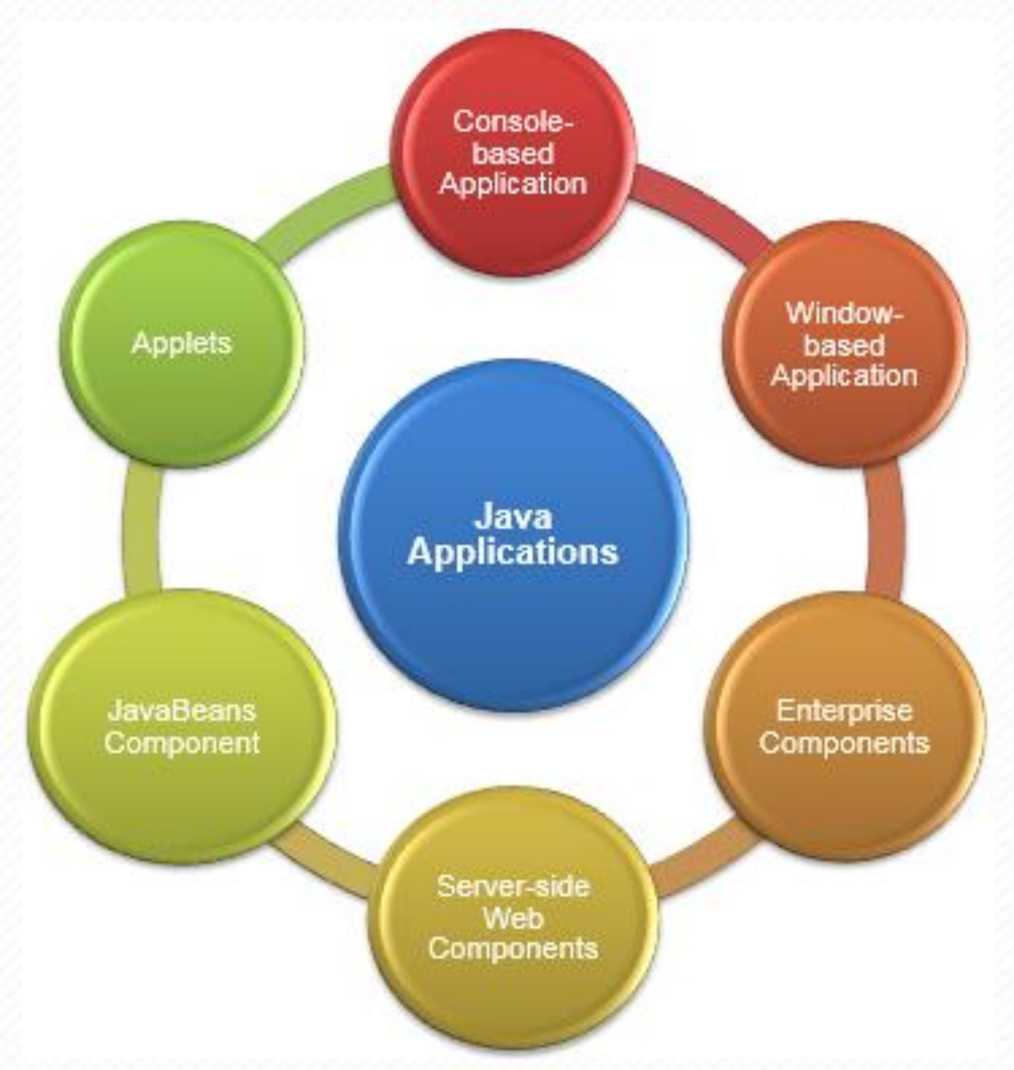


Basic Java Language

Tools and APIs	java	javac	javadoc	jar	javap	JPDA
	JConsole	Java VisualVM	Java DB	Security	Int'l	RMI
	IDL	Deploy	Monitoring	Troubleshoot	Scripting	Web Services
Deployment	Java Web Start		Applet/Java Plug-in			
User Interface Toolkits	Swing	Java 2D	AWT	Accessibility		
	Drag and Drop	Input Methods	Image I/O	Sound		
Integration Libraries	IDL	JDBC	JNDI	RMI	RMI/IIOP	Scripting
Other Base Libraries	Beans	Int'l Support	Input/Output	JMX		
	JNI	Math	Networking	Override Mechanism		
	Security	Serialization	Extension Mechanism	XML JAXP		
Base Libraries (lang and util)	lang and util	Collections	Concurrency Utilities	JAR		
	Logging	Management	Preferences API	Ref Objects		
	Reflection	Regular Expressions	Versioning	Zip	Instrumentation	

Java HotSpot Client and Server Virtual Machine

Java applications



Java IDE

- You can open a basic text editor like notepad and start writing basic Java programs or continue with the integrated development environment (IDE).
- Some Java IDE:
  - NetBeans,
  - Eclipse Java IDE,
  - JDeveloper,
  - Android Studio,
  - MyEclipse
  - JEdit.
  - IntelliJ IDEA
  - Visual Studio Code-Plugin Java

Hello, world

```
import java.io.*;
public class HelloWorld
{
    // Your program begins with a call to main().
    public static void main(String args[])
    {
            /*Prints "Hello, World" to the terminal window. */
            System.out.println("Hello, World");
    }
}
```

Hello, world

- Line 1 **import java.io.*** is a library declaration command.
- In Java, every line of code that can actually run needs to be inside a class. Line 2 **public class HelloWorld** declares a class named HelloWorld, which is public, that means that any other class can access it.
- Line 4 or 7 **//… or /*…*/** will be ignored by the compiler and it is called comments in the program.
- Line 5 **public static void main(String[] args)** this is the entry point of a Java program.
- Line 8 **System.out.println("Hello, World")** is a method that can be used to print a line to the screen, it is a part of System class that is a pre-defined class in Java and it holds some useful methods and variables.

Basic syntax

- Tokens:
  - A Java program consists of various.
  - A token is either a keyword, an identifier, a constant, a string literal or a symbol.
- Semicolons:
  - The semicolon (;) is a statement terminator.
  - Each individual statement must be ended with a semicolon.
  - It indicates the end of one logical entity.
- Comments:
  - Comments are like helping text in java program & they are ignored by the compiler.
  - Examples: // Your program begins with a call to main().
    /*Prints "Hello, World" to the terminal window. */

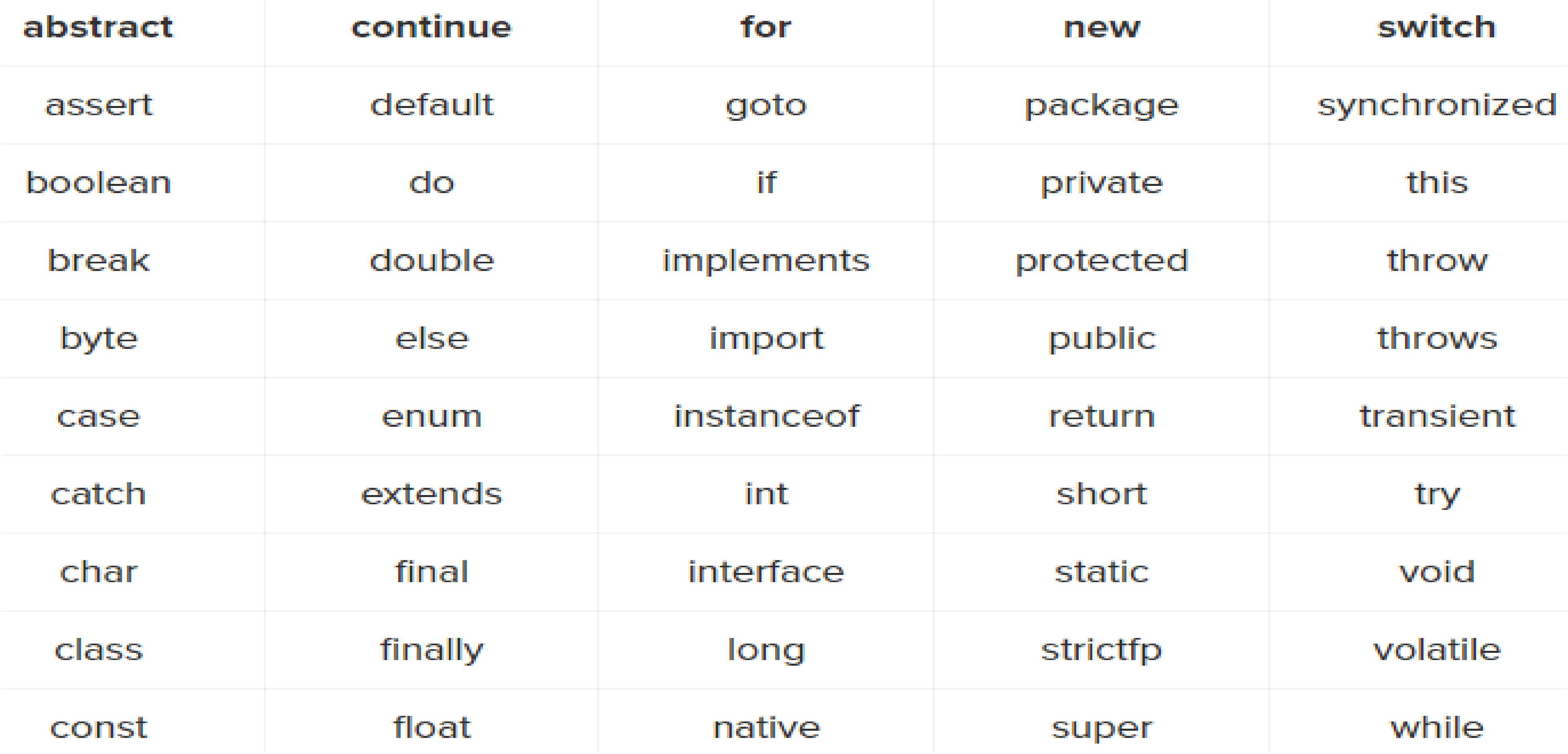
Basic syntax

- Identifiers:
  - Identifier is a concept used to name a variable, constant, function, class, ... in the program
  - It is conventionally named as follows:
    - start with a letter or _
    - do not contain special characters such as @, #, ...
    - cannot match keywords in java language
  - Java is a case-sensitive programming language.
  - Examples:

myName	ivalue	_value	retVal
add123	j	a_b_c	T

Basic syntax

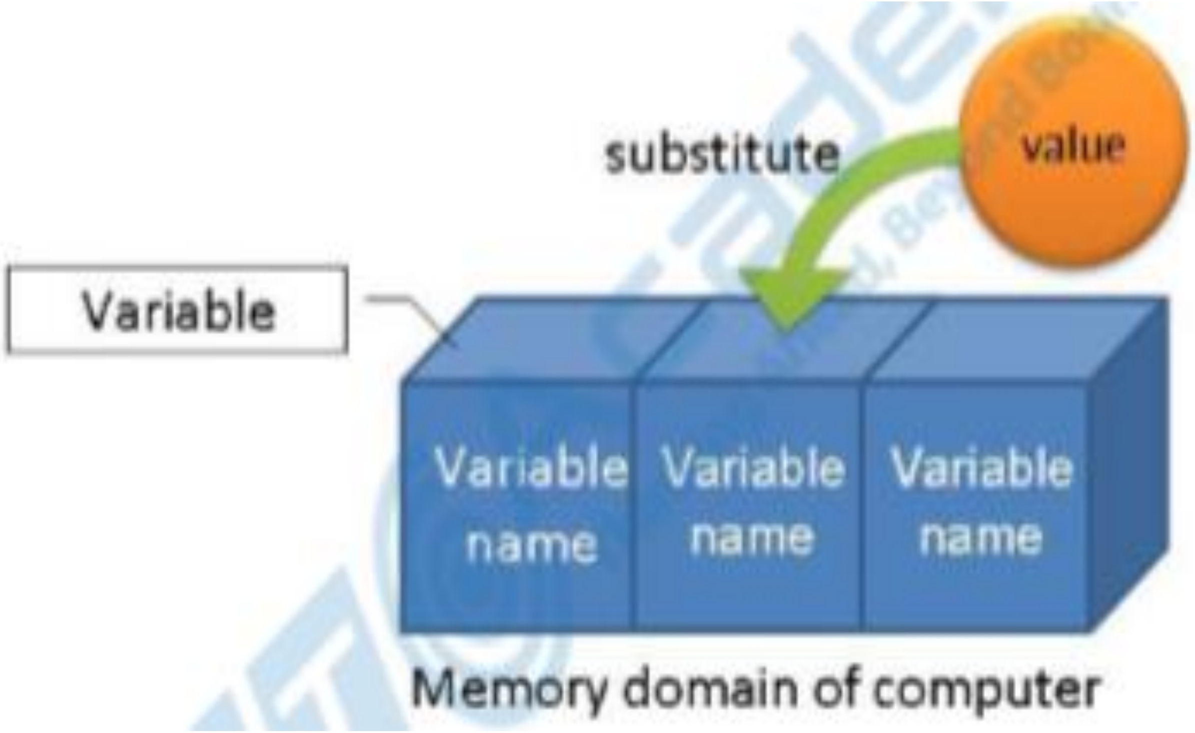
- Keywords: The following list shows the reserved words in Java. Keywords in java:



abstract	continue	for	new	switch
assert	default	goto	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while

Variable

- Helps the system create memory areas to store values in the application.
- Each variable has its own data type, which varies in size depending on the *data type* - indicating the ability to store the value of the variable.
- Each variable must also be given a *name / identifier* to help the system manage and access data in its memory.



Variable declaration

- Syntax variable declaration in java

  <data type> <variable_name> [= value];

- Examples:
  - double salary;
  - int count = 0;
  - boolean done = false;
  - long earthPopulation;

Constant

- Constants are similar to variables, but especially if a variable is declared as a constant, it will not be changed during the program.
- To declare a constant in JAVA, we add final before the variable declaration syntax and must assign it a value..

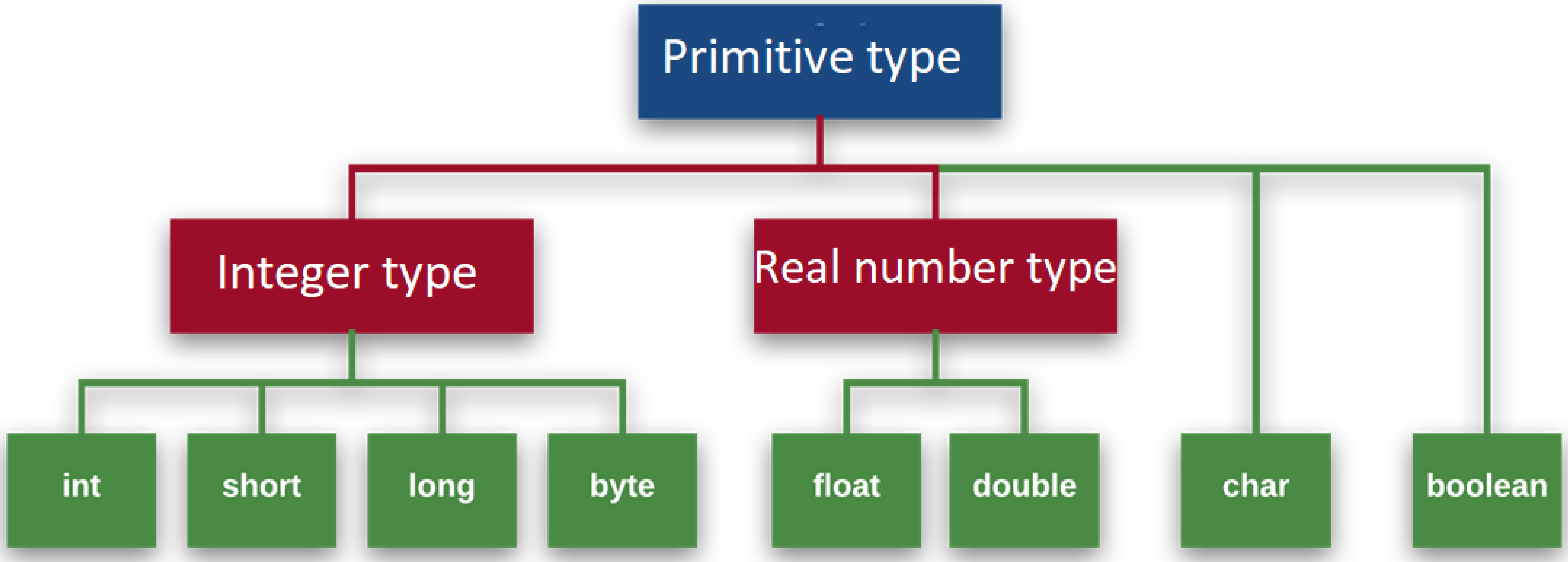
<final> <data type> <constant_name = value>;

- Examples:
  - final float PI = 3.14f;
  - final char FIRST_CHARACTER = 'a';
  - final int VIP_TYPE = 1;

Data type

- The data type tells the system how big a variable / constant is, this magnitude indicates the ability to store the value of the variable.
- Choosing a data type for a variable depends on the problem to be solved. Usually we should rely on the function or ability to store the value of that variable.
- Data types in JAVA include: “primitive” type and “extended/reference” type.

Data type – primitive type



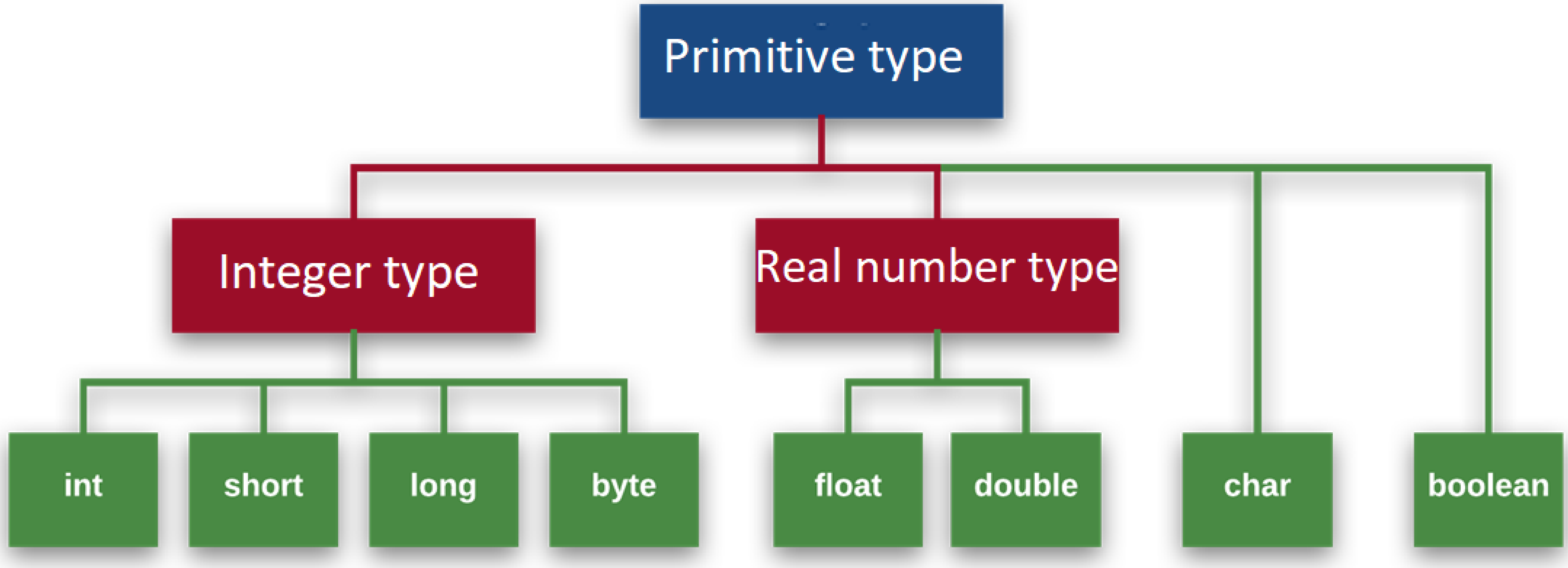
Data type

- The data type tells the system how big a variable / constant is, this magnitude indicates the ability to store the value of the variable.
- Choosing a data type for a variable depends on the problem to be solved. Usually we should rely on the function or ability to store the value of that variable.
- Data types in JAVA include: “primitive” type and “extended/reference” type.

Data type

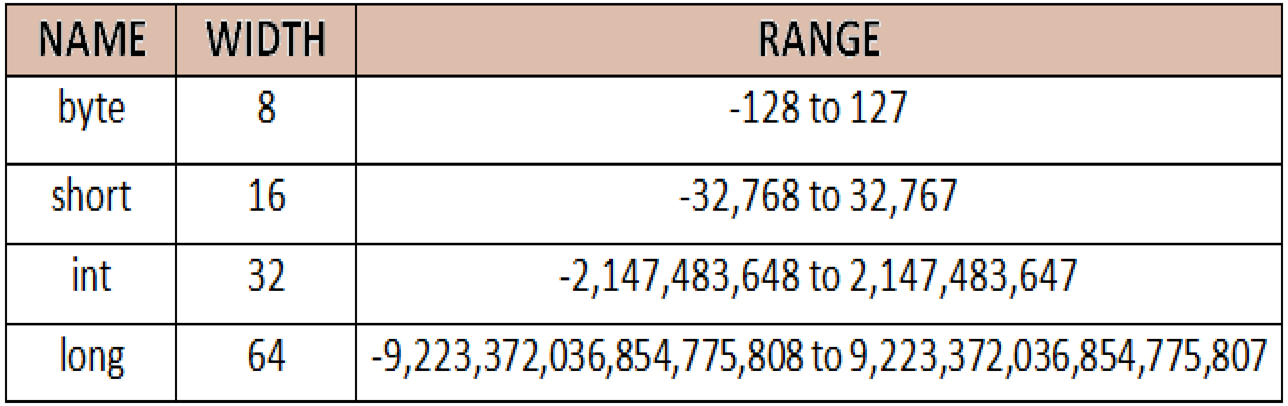
- The data type tells the system how big a variable / constant is, this magnitude indicates the ability to store the value of the variable.
- Choosing a data type for a variable depends on the problem to be solved. Usually we should rely on the function or ability to store the value of that variable.
- Data types in JAVA include: “primitive” type and “extended/reference” type.

Data type – primitive type



Data type – integer type

- For example:
  - correct:
    - byte month = 5;
    - short salaryUSD = 2000;
  - Incorrect:
    - byte day = 365;
    - short salaryVND = 40000000;



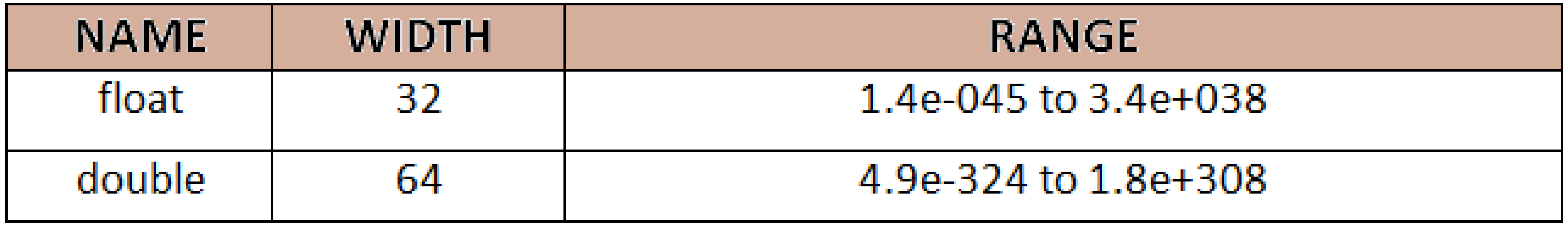
NAME	WIDTH	RANGE
byte	8	-128 to 127
short	16	-32,768 to 32,767
int	32	-2,147,483,648 to 2,147,483,647
long	64	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807

Codingeek.com

- Note: If no value is declared for the variable, the integer variable type will default to 0 (or 0L for long type).

Data type – real number type

- This type is used to store and calculate real numbers, which are floating-point numbers, including:



NAME	WIDTH	RANGE
float	32	1.4e-045 to 3.4e+038
double	64	4.9e-324 to 1.8e+308

Codingeek.com

- For example:
  - float rating = 3.5f;
  - double radius = 34.162;
- Note: If no value is declared for the variable, then the actual numeric variable has a default value of 0.0f for the float type and 0.0d for the double type.

Data type – char type

- The char type has a width of 1 byte, used to store a character enclosed in single quotes ('') like: 'a', 'B' ...
- For example:
  - Correct:
    - char thisChar = 'a';
    - char gender = ‘M ';
  - Incorrect:
    - char thisCharFail1 = "a";
    - char thisCharFail2 = b;
    - char thisCharFail3 = 'ab';

Data type – boolean type

- The boolean type in Java is only represented by two values: true and false.
- In java, it is used in checking logical conditions, not in calculations.
- For example:
  - correct: boolean male = true, graduated = false;
  - incorrect: boolean male = 1, graduated = 0;
- Note: If no value is declared for the variable, then the boolean variable will have a default value of false.

Data type – reference type

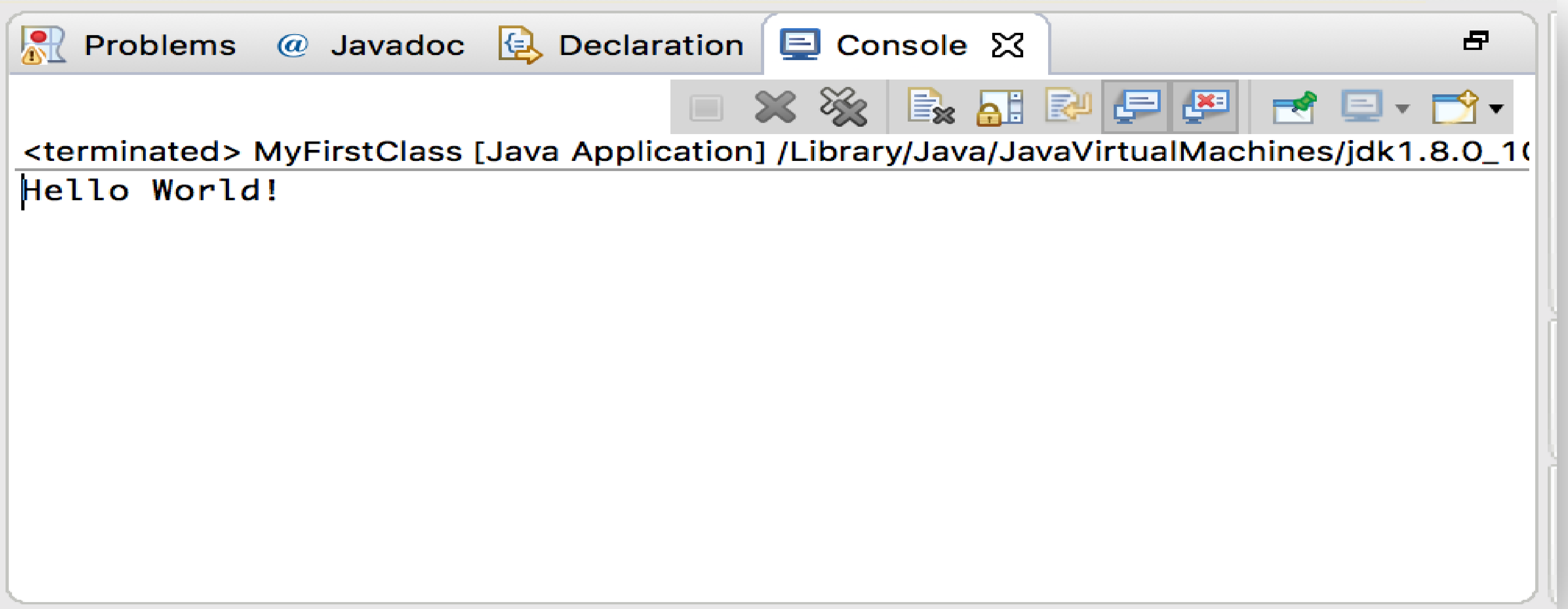
- In Java, the type created by combining primitive types together called reference types, usually created by classes (will learn more in OOP module).
- The reference type created by JAVA and most commonly used is String – it is a combination of character and array type.
- Illustrative image: String sayHello = "HELLO"; =>

H	E	L	L	O

- For example:
  - String myName = "My name’s Phong";
  - String myAddress = "184 Le Dai Hanh";
  - String str = "";

Console Input/output

- Console as a dedicated console, in programming languages, Console is known as a simple application control through text lines (command line), it is distinguished from UI control.



```
<terminated> MyFirstClass [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_1(
Hello World!
```

Console Input

- There are several ways to Input on Console in JAVA, here we only learn how to use Scanner class in the library is package java.util.Scanner
- To import, we need to create 1 Scanner object as the following example:

```
Scanner scanIn = new Scanner (System.in);
```

- Some commonly used methods for input on Console:

• **next (), nextLine ()**	enter the string
• **nextByte (), nextInt (), nextLong ()**	enter integers of type byte, int, long
• **nextFloat (), nextDouble ()**	enter a real number of type float, double
• **nextBoolean ()**	enter true / false type (Boolean)
• …	

- We will learn in detail through the last example.

Console output

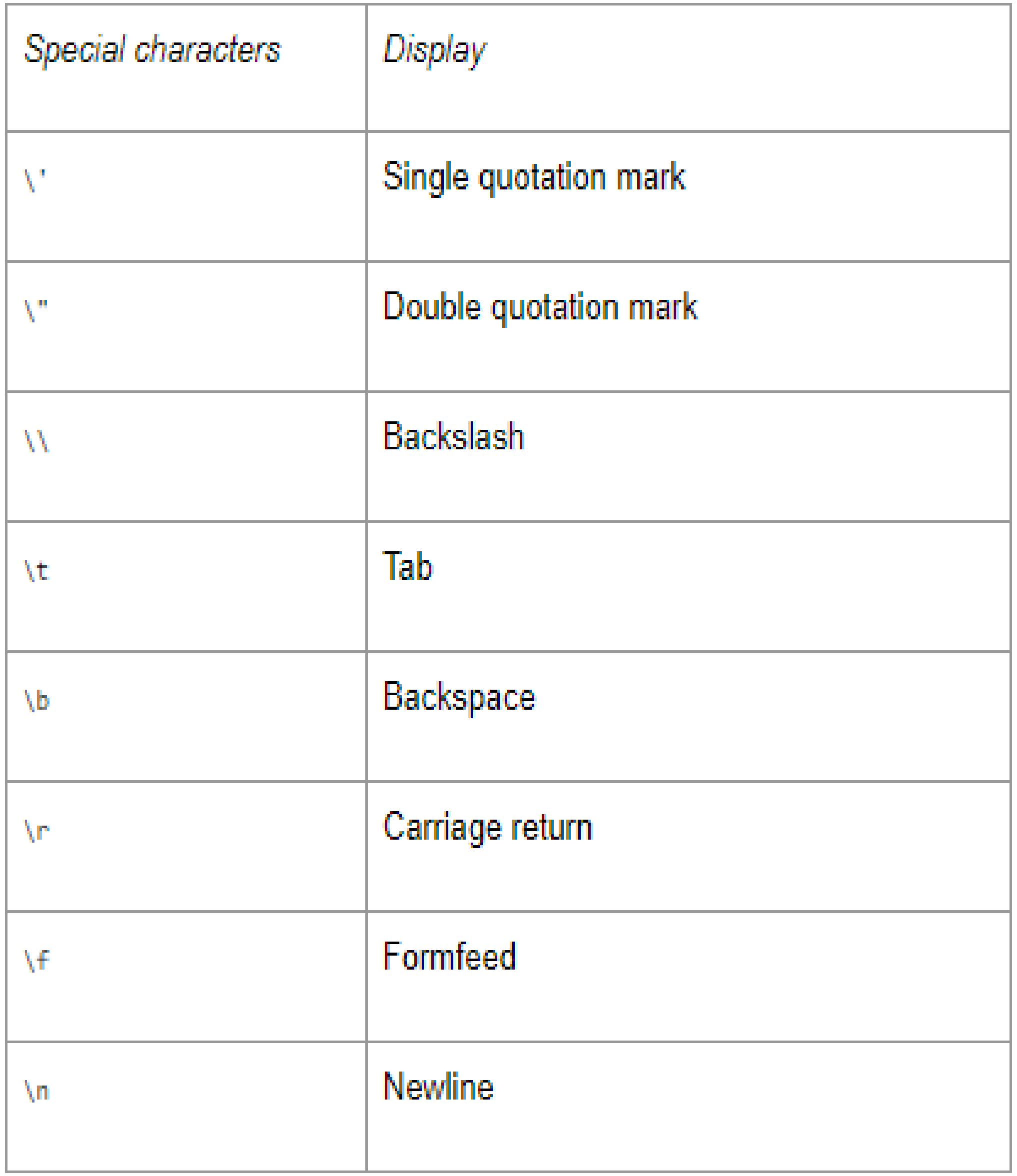
- Output with line breaks:

System.out.println (Format string);

- Output without line breaks:

System.out.print (Format string)

- Format string: is what needs to be output to the screen.
- You can insert several special characters into **Format string**. The side table shows these special characters; note that each is preceded by a backslash (\).
- The java string format() method returns the formatted string by given locale, format and arguments. If you don't specify the locale in String.format() method, it uses default locale by calling Locale.getDefault() method. The format() method of java language is like sprintf() function in c language and printf() method of java language.



Special characters	*Display*
\'	Single quotation mark
\"	Double quotation mark
\\	Backslash
\t	Tab
\b	Backspace
\r	Carriage return
\f	Formfeed
\n	Newline

Java String format()

- Syntax: String.format(String format, Object... args): return a formatted string. Below, that is a table of format specifiers supported by the Java String.

Symbol	Data Type	Output
%a	floating point (except BigDecimal)	Returns Hex output of floating point number.
%b	Any type	"true" if non-null, "false" if null
%c	character	Unicode character
%d	integer (incl. byte, short, int, long, bigint)	Decimal Integer
%e	floating point	decimal number in scientific notation
%f	floating point	decimal number
%g	floating point	decimal number, possibly in scientific notation depending on the precision and value.
%h	any type	Hex String of value from hashCode() method.
%n	none	Platform-specific line separator.
%o	integer (incl. byte, short, int, long, bigint)	Octal number
%s	any type	String value
%t	Date/Time (incl. long, Calendar, Date and TemporalAccessor)	%t is the prefix for Date/Time conversions. More formatting flags are needed after this. See Date/Time conversion below.
%x	integer (incl. byte, short, int, long, bigint)	Hex string.

Console output

- Example:

```
String name = “Phong”;
char gender = “M”;
int age = 33;
```

- Vertical outputing:

```
System.out.println("Name: "+name);
System.out.println("Gender: "+gender);
System.out.println("Age: "+age);
```

- Horizontal outputing:

```
System.out.println("Name: "+name+"\tGender: "+gender+ "\tAge: "+age);
//or use String.format()
System.out.println(String.format("Name: %s \tGender: %c\tAge: ",+name,gender,age));
```

Console Input/output example

```
import java.util.Scanner;
public class ScannerDemo1 {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in); // declare Scanner object
        System.out.print ("Input your Name:");
        String name = sc.nextLine(); // enter string
        System.out.print ("Input your gender (F / M):");
        char gender = sc.next().charAt(0); // enter the character
        System.out.print ("Input your age:");
        int age = sc.nextInt(); // enter int int
        System.out.print ("Input your phone no:");
        long mobileNo = sc.nextLong(); // enter integer long
        System.out.print ("Input your CGPA:");
        double cgpa = sc.nextDouble(); // enter real double number
```

Console Input/output example

```
        // output values
        System.out.println("Name: "+name);
        System.out.println("Gender: "+gender);
        System.out.println("Age: "+age);
        System.out.println("Mobile Number: "+mobileNo);
        System.out.println("CGPA: "+cgpa);
    }
}
```

References

- https://itviec.com/blog/hoc-lap-trinh-java/
- https://gpcoder.com/1638-tong-quan-ve-ngon-ngu-lap-trinh-java/
- https://www.learnjavaonline.org/en/
- https://www.codingeek.com/java/primitive-data-types-in-java-integers-floating-point-character-and-boolean/
- http://www.informit.com/articles/article.aspx?p=30241&seqNum=3
- https://yellowcodebooks.com/2016/11/01/java-bai-7-nhapxuat-tren-console/
- https://www.geeksforgeeks.org/scanner-class-in-java/
- https://www.javatpoint.com/java-string-format

VTC Academy

Think Ahead, Beyond Boundaries

Thank you!