UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA-MINTIC
LANGUAGE DEPARTMENT

BASICS OF JAVA

INTRODUCTION TO THE JAVA PROGRAMMING LANGUAGE

Introduction to Java Programming Language

Java is a general-purpose, **object-oriented programming language** designed for the development of software for consumer electronic devices, such as TVs, VCRs, toasters, etc.

Java is a platform neutral language, which means that **it is not tied to any particular hardware or operating system**. It guarantees users to 'write once, run anywhere.' The Java language **is supported by** almost every operating system, such as **Sun Solaris, RedHat, Windows**, etc.

The Java programming language was developed by **Sun Microsystems** of the USA in **1991**, it was originally called **Oak** by **James Gosling**, who was one of the inventors of the language. The **main goal** for the developers was to make the language **highly reliable, portable and simple**.

The team for the development of Java language included **Patrick Naughton**, who discovered that the existing languages such as C and C++ had some major drawbacks in terms of reliability and portability. They modeled the new language Java on C and C++, while removing some features that they considered **constraints**. This made Java a really simple, portable, and powerful language. To learn Java basics, let's revise the Java syntax.

The History of Java

There is a chronicle of events that occurred during the course of **development** of the Java language.

1990: A team of Sun Microsystems programmers decided to develop a special software to manipulate consumer electronic devices. The team headed by James Gosling.
1991: The team studied various languages present at that time, viz., C, and C++, and announced the new language to be "Oak."
1992: The team at Sun, known as **the Green Project**, demonstrated the application of their new language, for example, to control a list of home appliances using a hand device with the tiny touchscreen.

1993: The World Wide Web came to the Internet and transformed the text-based Internet into a graphical, rich environment. The Green Project team came up with an idea of developing **Web applets** (tiny programs) that could run on all types of computers connected to the Internet.

1994: The team developed **a web browser** called "**HotJava**" to locate and run applet programs on the Internet. This made it immensely popular amongst Internet users.

1995: **Oak** was renamed **as "Java,"** due to some legal snags. Java is the name and not an acronym.

1996: Java programming was established as the leader for Internet programming and also as a general-purpose, **object-oriented programming language**. Sun released the Java Development Kit 1.0.

1997: Sun releases the Java Development Kit 1.1 (JDK 1.1)

1998: Sun releases Java 2 with version 1.2 of the Software Development Kit (SD K 1.2)
1999: Sun releases the Java 2 platform, Standard Edition (J2SE), and Enterprise Edition (J2EE)

2000: Sun releases J2SE with SDK 1.3.
2002: Sun releases J2SE with SDK 1.4.

2004: This marked the release of J2SE with JDK 5.0 (instead of JDK 1.5), known as J2SE 5.0.
2006- Sun releases Java SE 6.

2011: Sun releases Java SE 7.
2014: Sun releases Java SE 8.
2017: Sun releases Java SE 9.

Features of Java Programming

The main goal was **to design a language that could offer solutions** to problems encountered in modern programming. The goal was for the language to be **reliable, portable, and distributed**, and at the same time, it needed to be **simple, compact, and interactive**.

Compiled and Interpreted

Java language combines both of these approaches, thus making Java a two-stage system. This approach was never offered before, as any language before was either compiled or interpreted.

Firstly, the **Java compiler translates source code into bytecode instructions**; bytecodes are not machine instructions.

Secondly**, the Java interpreter generates machine code** that can be directly executed by the machine that is running the Java program.

Independent and Portable

Java programs **can be easily moved from one system to another**, anywhere and anytime. With changes or an upgrade in the operating system, processors and system resources will not force any changes in Java programs.

Java programming ensures portability in two ways. Firstly**, Java compiler generates bytecode instructions** that can be implemented on any machine, and secondly, **the size of the primitive data** types are **machine independent.**

Object-Oriented

Almost everything in Java language is an object, which makes it a true object-oriented language. **All program code and data reside within objects and classes**. Java comes with an extensive set of classes that are arranged in packages, which can be used in program inheritance.

Robust and Secure

Java **provides many safeguards to ensure reliable code**. It has strict run-time, checking for data types. It is designed like a garbage collected language, i.e., it **captures series errors and eliminates any risk** of crashing the system.

Java systems verify all the memory access and, thus, ensure that **no virus is communicated** with an applet.

Distributed

Java programming **facilitates both the sharing of data and programs**. Java applications can open and access remote objects on the Internet as easily as on any local system.

Simple, Small, and Familiar

Java is a simplified version of C++, which is why it is familiar and yet different as it eliminates all the redundant and unreliable code. For example, Java does not use pointers, preprocessor header files, and many others. It also eliminates operator overloading and multiple inheritances in Java.

Multithreaded and Interactive

Multithreaded means **handling different tasks simultaneously**. Java language supports multithreaded programs, which means that we need not have to wait for one task to finish for another to start. This feature of Java greatly improves the interactive performance of graphical applications.

High Performance

Java programming performance is very impressive, considering the fact that is an interpreted language, mainly because of the bytecodes. Java architecture is **designed to reduce overheads**.

Dynamic and Extensible

Java is a dynamic language; **it is capable of dynamically linking in new class libraries, methods, and objects**. It can also determine the type of class through a query.

Ease of Development

Java 2 standard edition (J2SE) 5.0 supports features such as **Generics, Enhanced for loop, Autoboxing or unboxing, Type-safe enums, varargs, Static import, and Annotation**. These Java features make it easy for Java programmers by shifting the responsibility of creating the reusable code to the compiler, and also, the resulting code is free from bugs.

Scalability and Performance

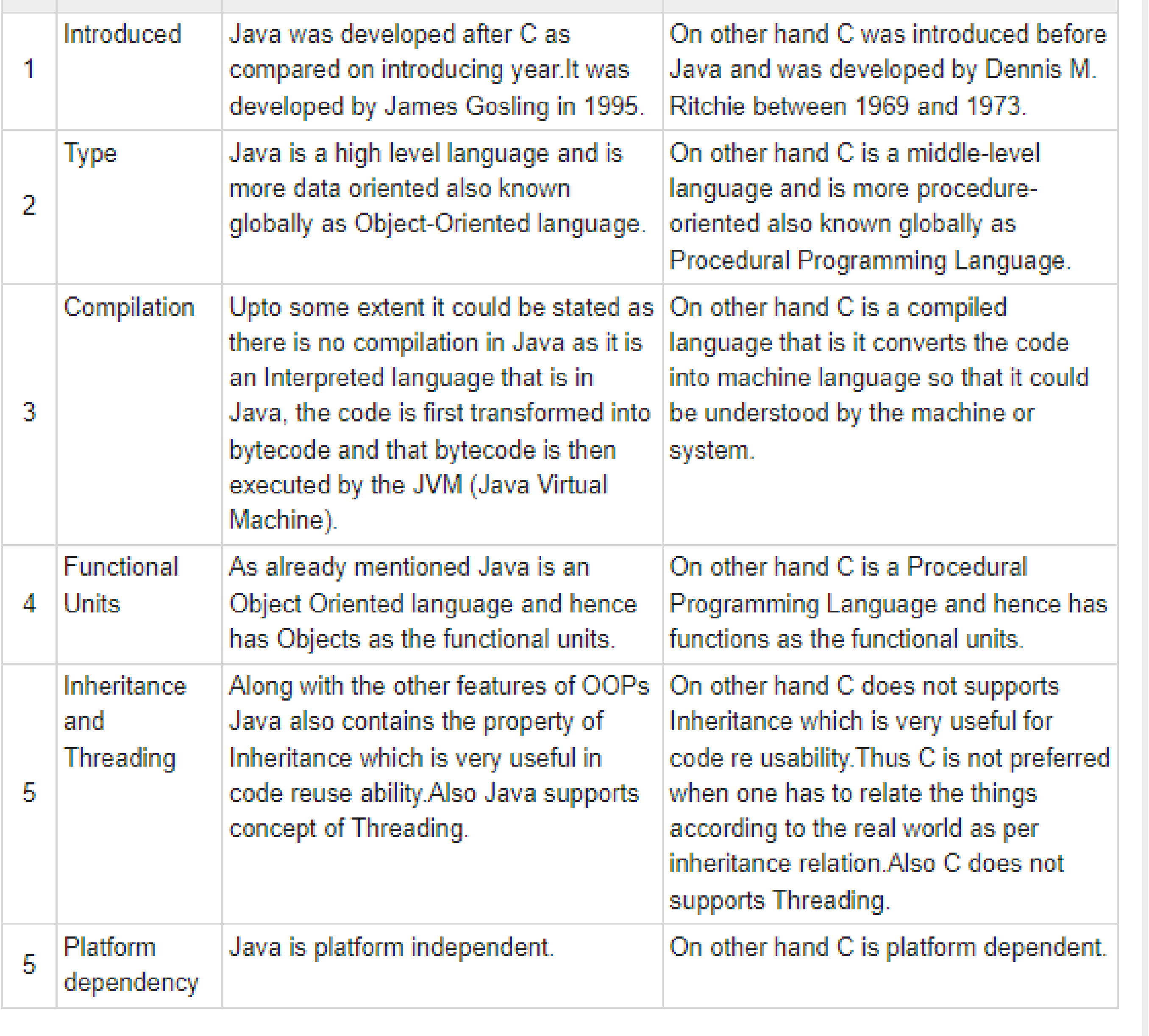
J2SE 5.0 **improves the startup time and reduces the amount of memory** used in the Java 2 runtime environment.

COMPARISON: C VS. JAVA VS. C++

The differences between these three programming languages are:

C Vs. Java

The major difference between C and Java is that Java is an object-oriented language and has a mechanism to define classes and objects.

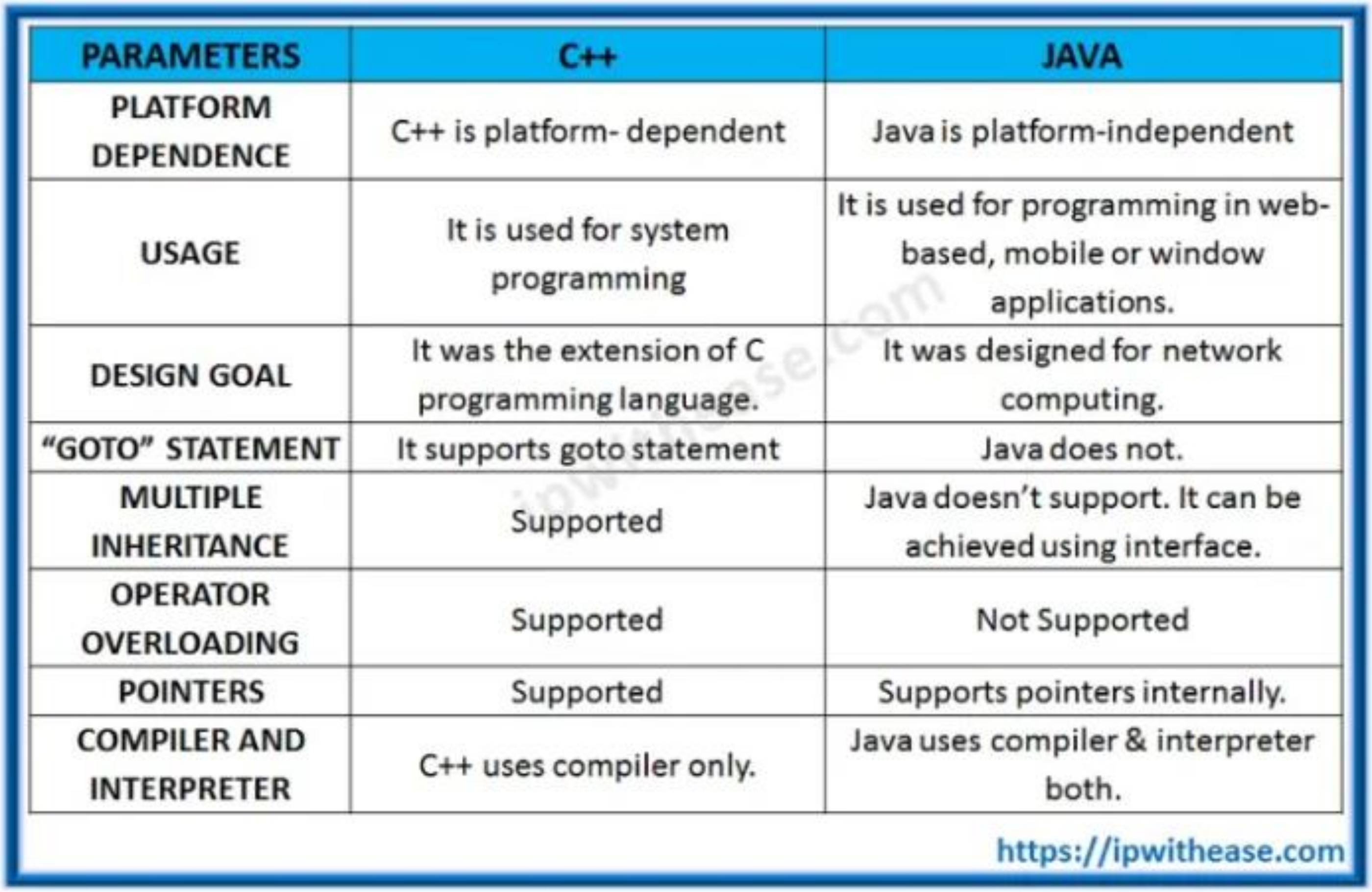


1	Introduced	Java was developed after C as compared on introducing year.It was developed by James Gosling in 1995.	On other hand C was introduced before Java and was developed by Dennis M. Ritchie between 1969 and 1973.
2	Type	Java is a high level language and is more data oriented also known globally as Object-Oriented language.	On other hand C is a middle-level language and is more procedure-oriented also known globally as Procedural Programming Language.
3	Compilation	Upto some extent it could be stated as there is no compilation in Java as it is an Interpreted language that is in Java, the code is first transformed into bytecode and that bytecode is then executed by the JVM (Java Virtual Machine).	On other hand C is a compiled language that is it converts the code into machine language so that it could be understood by the machine or system.
4	Functional Units	As already mentioned Java is an Object Oriented language and hence has Objects as the functional units.	On other hand C is a Procedural Programming Language and hence has functions as the functional units.
5	Inheritance and Threading	Along with the other features of OOPs Java also contains the property of Inheritance which is very useful in code reuse ability.Also Java supports concept of Threading.	On other hand C does not supports Inheritance which is very useful for code re usability.Thus C is not preferred when one has to relate the things according to the real world as per inheritance relation.Also C does not supports Threading.
5	Platform dependency	Java is platform independent.	On other hand C is platform dependent.

taken from: https://www.tutorialspoint.com/difference-between-java-and-c-language

Java Vs. C++

The main difference between C++ and Java is that Java is a true object-oriented language while C++ just adds an object-oriented extension to C. The increment operator in C++ indicates the same thing.



PARAMETERS	C++	JAVA
PLATFORM DEPENDENCE	C++ is platform- dependent	Java is platform-independent
USAGE	It is used for system programming	It is used for programming in web-based, mobile or window applications.
DESIGN GOAL	It was the extension of C programming language.	It was designed for network computing.
"GOTO" STATEMENT	It supports goto statement	Java does not.
MULTIPLE INHERITANCE	Supported	Java doesn't support. It can be achieved using interface.
OPERATOR OVERLOADING	Supported	Not Supported
POINTERS	Supported	Supports pointers internally.
COMPILER AND INTERPRETER	C++ uses compiler only.	Java uses compiler & interpreter both.

https://ipwithease.com

taken from: https://ipwithease.com/c-vs-java/