

# Comparative Analysis of C and Java Programming Languages

## Introduction

Learning a programming language is like learning a new foreign language; however, this language will be understood by the computer. Programming languages are tools for converting ideas about solving a problem into instructions that can be processed by the machine. Whether this is a high-level language or a low-level language, eventually human-written code will be interpreted/compiled into machine language and run on the processor. We learned the C programming language in the introduction to programming. After being able to write code fluently in C, we worked on C++ and Java. Although these languages are close, the mechanics here are similar to learning a new spoken language. Again, like human languages, some of the programming languages are very similar to each other, while others are completely different.

A programming language is an artificial language designed to demonstrate various calculations with machines, mostly computers. programming languages have made an important development starting in the 1950s. As a result, hundreds of languages were invented. With the development of technology, it has become possible to make more diverse applications with more advanced hardware and fast processors. These applications demand more efficient and powerful programming languages.

Written by Dennis Ritchie, the "C" was introduced in 1972 at the US AT&T Bell laboratories. Again in this laboratory, it was named C because it is the predecessor of the B language developed before it. This language was created to develop the widely used UNIX operating system. While designing C, it was aimed to be useful and to focus on the work of programmers, in this way

productivity will be ensured. As a result of this design, since it is more efficient than other languages, it has started to be used by more people, out of Bell. And in the next 10 years, it took its dominant place among programming languages. A committee of ANSI (American National Standards Institute) standardized C in 1989 and published the version called "ANSI C". The vast majority of today's C compilers support this standard. This standard was later adopted by ISO (International Standards Organization).

Java was developed in 1991 by James Gosling of Sun Microsystems and his team. Java was originally designed for programming home appliances. A few years later, James Gosling began to connect Java with the internet. By 1995, the Netscape company released an update that could run Java programs in the Netscape browser. Java's old name was Oak, but it was used by another programming language. After the obligation arising from this, the name was changed to Java, inspired by a coffee bean. Although Java is seen as a programming language for internet applications as it was in its first use, it is actually a general purpose language that can be used independently.

C (like the letter c) is a general-purpose computer programming language. It continues to be widely used and effective. By design, C's capabilities are evident in low-level programs. It has found permanent use in operating systems, device drivers, networking protocols, but application development with C has waned. C is widely used in computer architectures of many sizes, ranging from microcontrollers to supercomputers. The popular programming language C, which was also used in the development of Linux, was originally developed to create utilities running on Unix. C was used when re-developing the kernel of the Unix operating system. It later became one of the most widely used programming languages, with C compilers available for operating systems running on nearly all modern computer architectures. C is an imperative language with a static type system that supports structured programming which will be discussed later on. It is designed for low-level access to memory and language constructs that are properly met with Machine instructions with minimal runtime support. Despite or thanks to its low-level capabilities, the language encourages multiplatform programming. A standard C program written with attention to portability can be compiled for a wide variety of computer platforms and operating systems with few changes to the source code. From 2000 to 2022, C has always been among the first two languages in the TIOBE index, which also shows that

it is a popular programming language, although it has been surpassed by Java and Python for a period of time.

Java is a general-purpose, high-level, object-oriented and class-based programming language. It aims to allow programmers to write once and run anywhere (Write Once Run Anywhere). It is designed to have as few application dependencies as possible. Compiled Java code can run on any platform that supports Java without the need to recompile. Java applications are typically compiled to bytecode that can run in a Java virtual machine (JVM). The JVM, on the other hand, can run on any computer architecture, so the compiled java bytecodes work with any architecture that is a JVM for it. Java's syntax is similar to C/C++, but has higher level possibilities than either. In contrast to conventional compiled languages, the Java runtime offers dynamic features (such reflection and runtime code modification). According to GitHub, which estimated 9 million developers as of 2019, Java was one of the most widely used programming languages primarily for client-server web applications. Java was created at Sun Microsystems by James Gosling and launched in May 1995. It was first made available as a core component of Sun Microsystems' Java platform. Sun previously issued specific licenses for Java compilers, virtual machines, and libraries. As of May 2007, Sun had re-licensed most Java technology solely under the GPL-2.0 license, in compliance with the Java Community Process requirements. Although Oracle provides its own Java Virtual Machine, the official one is the OpenJDK JVM, which is free open source software that is the default JVM for Linux distributions. Mostly developers are using OpenJDK and Java 19 is the most recent release as of September 2022, with Java 17, 11, and 8 serving as the current long-term support (LTS) releases.

There are several paradigms that are mentioned in the definition of C and Java. Most programming languages do not consist of just one paradigm. If it occurs, it is called pure. Pure languages like Pure OOP and Pure Functional are very rare. Most languages contain more than one paradigm. For example C is imperative and structured, Java is object-oriented, functional and imperative. These languages are called multiparadigm.

- Procedural(Imperative) programming is a paradigm that uses a clear set of instructions to modify the state. It has the advantage of being close to the machine, so it is efficient. Also

imperative languages like C are popular. Therefore programmers are more familiar with this type of languages. But the semantics can be complex and hard to understand due to side effects. Thus creates a higher learning-curve for newcomers. These facts also make programs hard to debug. And abstraction is more limited.

- Object-oriented programming is creating objects that communicate with one another. Public interfaces and their own internal (encapsulated) state are features of objects. Can be class-based and prototype-based. The OOP paradigm's most distinctive quality is its support for inheritance. OOP's main advantage over other programming paradigms is inheritance, which makes it possible to reuse and extend code without having to modify the original source code. A high degree of modularity is provided by the method of representing a program as a collection of objects of different classes, and moreover by specifying many classes as extensions or changes of other classes.
- Generic programming is a method of computer programming involving writing algorithms in terms of later-specified types that are then instantiated as necessary for particular types that are supplied as inputs.
- Structured programming is rather of using gotos to establish control flow, instead uses nested loops, conditionals, and subroutines. Typically, blocks have local variables (have lexical scope).
- Functional programming is using function calls in programming to prevent any global state.

## General Comparison

First of all, the same things between C and Java will be mentioned. Value, type and expressions in Java are similar to C. Variables, conditions (if, switch), loops except for each, function call, return and argument structures, primitive types, standard libraries are also the same.

There are certain significant areas in which C is different from Java. In C, classes/objects not included and not object-oriented. Instead it has structured types such as struct and union. We have ways to do inheritance and abstraction. Also C can add new types with typedef to its type system. Arrays are not aware about bounds and size, they are simpler. Strings are the same. No collections

classes and exception handling. Programmer manages the memory ownself. With explicitly allocating and freeing the memory for dynamic data structures. For example linked lists, binary trees and so forth. With pointers memory access is more powerful but also dangerous due to machine-close structure of C. This allows it to be more flexible but less reliable.

## Syntax and Semantics

To take a closer look at the C programming language, comments are very important for writing clean and understandable program. The values that the program computes are literal. The comment syntax comes from C to Java. In Java and C `/* */` used for multiline comments that ignored by the compiler. Modern C also recognizes `//` for single line comments. Primitive numeric types are comes from C to Java, that can seen from table. Int, short, long, float, double, char are same. Byte and boolean types are only included in Java. Historically these types were specified with kindly lower perfection than in Java. For illustration, int was whatever was most natural for a given machine, and could be 16 bits, 32 bits, or commodity differently. I'm not going to dwell on these details. However, you can lookup how to do that, If you need to know how numerous bits are in an int. Two of Java's primitive types don't live in C byte and boolean. That's ok. utmost people do n't use byte anyway. In C, use an int( or a short if you 're really concerned about memory operation) and just do n't assign it a value lower than-128 or lesser than 127. I told you C gave you less protection than Java. Now booleans are authentically useful, but if you suppose about it, they simply represent a true- false, yes- no, 1- 0 value. So again, in C, use an int. C uses the convention that 0 means " false " and any non-zero value means " true. " Boolean variables and functions that return Boolean values are traditionally declared int. The stdbool library provides support for further unequivocal booleans in ultramodern C compilers.

In Java, a housekeeper represents a 16- bit Unicode character( a " law point "). In C, a housekeeper is an 8- bit byte representing an ASCII character. The difference matters if you 're working with ultramodern textbook sources which can include foreign characters, accentuations, calculation symbols, and, yes, emoji. There's some library support for handling unicode in C, but it isn't a strength of the language. On the other hand, a C housekeeper is exactly a byte, so you could use it for that if you wanted(there are some inked/ unsigned calculation considerations, and C has

unsigned numeric types, including unsigned housekeeper). Literals for primitive types are just like in Java. Numeric representations for integer and floating point figures. Character(housekeeper) literals in single- quotations. String literals enclosed in double- quotations.

Priority rules for the drivers are as in Java, and hiatuses can be used to group sub-expressions. In C there's no string concatenation using the driver. Strings aren't primitive. Java took "==" for "equals" comparison from C. Pascal( which came after C), used "=" for assignment and plain "=" for "equals", which actually makes further sense, but unfortunately no way caught on.

Similar to Java, variables in C must be defined before being used. In C, assigning a value to a variable is done using an assignment statement and, if the variable is a literal, the "=" (single equals sign) operator. This is similar to how it is done in Java. An "evaluable at build time" requirement must be met. Arrays are denoted by square brackets ([]) in both languages. An array variable is declared by appending [] to a type. In C, you must know the size of the array at compile time and use square brackets after the variable name.

Strings are objects in the Java programming language containing constructors, methods, and runtime verification. Through the string literal syntax and the overloading of "+" to imply string concatenation, Java additionally offers particular support for strings. Strings in C are just char arrays.

The condition must be enclosed in parenthesis in C, just like it does in Java, and if it is true, the block will be run. because there is only the zero/non-zero convention and no true boolean type. Iteration statements in Java are borrowed from those in C. The while clause, in both its do-while and other forms, is the same. For usage in iteration constructions, C provides the break and continue statements. These operate in the same manner as Java, ending the loop or immediately beginning the following iteration. Java's return statement and C's return statement are both used to extract a value from a function. In both, arguments to functions are passed by value. Before they may be utilized, C functions need to be defined. The order in which methods are defined in a class is irrelevant in Java. With the functions and structured types at hand, one of the few really important differences between C and Java is memory management. This concerns how C programs manage the memory they use to store values. This includes reference types, also known as "pointers", and how you use them to develop complex, dynamic data structures.

## Conclusion

Because it could be easily compiled by a simplistic compiler, C is a minimalistic programming language. C provides pointers for low-level memory access as well as direct hardware address access. For each of its fundamental language components, C generates a small number of machine language instructions, thus it doesn't need a lot of run-time assistance. It may be said that the C programming language is appropriate for a wide range of systems programming applications that were previously implemented in assembly languages. However, because C is a structured oriented programming language and emphasizes the procedural programming paradigm, controlling the large-scale program might be challenging. Java is the sole object-oriented programming language. It makes modular applications available to permit code reuse. Java is open source. Users can use it for free. The fact that Java is platform-independent is one of its most significant advantages. Programs built in Java may easily be moved across computer platforms. Java also has several shortcomings. Java is a memory-intensive computer language. Java is slower because there is an extra layer between the systems and the programs. The extra layer is the Java Virtual Machine (JVM). All actions performed by Java programs must be performed by the Java Virtual Machine. The system is compelled to comply with the right instructions after that.