Java vs C

In the TIOBE index for September 2021, we can see that Java is the 2nd and C is the 3rd. Two languages both are being used commonly nowadays as in the past. C programming language has procedural programming concept while Java has object-oriented programming concept. It should be started by explaining the difference between object-oriented programming and procedural programming paradigms.

Comparing by Programming Paradigm

Procedural programming is a programming paradigm. The main feature of this paradigm is that the program is a series of instructions that will be executed sequentially one after the other. Procedural programming languages focus heavily on splitting up programs into pieces called functions. The basic programming unit is function in this paradigm.

On the other hand, object-oriented programming is a programming paradigm that consist of class and object concepts. The main property of this paradigm is splitting the problem into small reusable pieces called classes or objects to struct a software. The basic programming unit is class in this paradigm.

This two programming paradigm is totally different. The most important difference is the maintainability support. In the Java programming that is an object-oriented language, there are concepts called inheritance, encapsulation, data hiding, polymorphism. These properties that provide security and class-object concept together support the reusability and maintainability properties. In the C language the focus is not on the maintainability and data. C has less security, it is hard to reuse the code, but it is easy to track flow of the program. The focus of C programming is efficiency.

Readability

After a while, coding was relegated to a different role from the one at the beginning. The maintenance became the main role of the code. And the readability has a major role for the maintenance. So, readability became an important measure of the quality of programs and programming languages. We can say that there is a big step from focus on machine to focus on human. Although the object-oriented structure of Java makes some parts of code less readable, Java is completely more readable than C. Because the C language is low-level language and this low makes the code more complex. Since the low-level languages is developed on machine oriented, they are less readable we can say. On the other hand, Java is human-oriented, it has inherently readability concerns. It is high-level language that is closer to our natural language. There are important two factors that affect the readability.

Orthogonality affects the readability. Orthogonality is an important property that makes the complex and big designs compact and readable. In an orthogonal design, operations do not have side effects; each operation only changes one situation without affecting others. The need for exceptions to a language's rules decreases with how orthogonal it is designed. For example, variables are passed by value, but arrays are passed by reference and an array can contain any data type. This example shows us an instance of lack of orthogonality in the C programming language. The accessibility specifiers public and private in Java language are totally unrelated to the static specifier. Variable types, however, determine how they are stored. So, there are side effects to specifying a variable's type. C considered as non-orthogonal language while Java programming language is Quasi-Orthogonal. Since the Java is more orthogonal, it makes it easier to read.

Overall simplicity that means having not large number of constructs to do some specific tasks affects the readability. We may learn different subset from the author's subset that is not familiar to us. One of design goals of Java language is simplicity. Moreover, the C programming language is medium or low level while Java high level. Java's overall simplicity is more coverage. Simplicity will be covered more detailed in the Writability paragraph.

We must compare the syntax of the C and Java. We can say that there are small differences. They have similar syntax. Control statements are literally the same in terms of syntax. Punctuations are the same. There are small differences as in tokens. It is called "small" differences because most of these differences are due to object-oriented structure of Java. Due to this object-oriented structure in Java, there are too many extra keywords such as public, private, extends, finally etc. There are also keywords for the extra data types such as Boolean and String. There is no Bool type and String type in C programming languages. It is an obligation to use integer or another data type to indicate true/false statements. This property of C programming language decreases the readability Similarly String type gives Java language a clean look for many operations while in C language you must deal with arrays. All the functions in a C source code are global while Java has namespace property. In the Java, methods and classes can be split to meaningful parts. It makes the code more readable and clearer.

It should be given some extra examples. Some basic operations such as printing, getting input, file operations are different since the Java is object-oriented programming and it needs to access some classes. Allocating memory syntax is different and this is a perfect example for comparing low level language and high-level language. Java has an extraordinary simplicity than C language in term of managing dynamic memory. This instance of simplicity that is not just in memory improves readability. Importing library syntax is different. Null keyword is different. The syntax of the main function is

different since the Java has object-oriented structure. Declaring constants are different (const vs final). Getting input is much easier than C in the Java in terms of readability since Java needs to access some classes. For loop initial declarations is not allowed in C while Java allows.

Writability

The definition of the writability is a measure of ease about how easily a programmer can code. Readability also affects the writability since if you cannot understand the code, you cannot develop it. It needs reread. Since the readability of the C programming language is less than Java, this is a criterion for us to evaluate their writability. Also, C is a low-level language. So, it has a complex structure. There are so many things that you must be careful while coding since you are in the deep. On the other hand, Java is high level language. It prevents a lot of things even not asking you. For instance, there will be no memory leak while you are coding Java. Using of pointers, also, makes things more confusing in the C programming language. There is no pointer in the Java, so many things are reference in the Java that makes coding easier. The string type of Java (that C programming language does not have) provides a clear writability advantage. While dealing with character arrays for string operations is a complex and tedious task, they can be easily handled in Java. Moreover, in Java language, being able to write functions using a design pattern called templates suitable for many data types not only relieves the programmer in terms of writability, but also brings the advantage of readability. Also, array lengths can be easily accessed in Java without using additional variables for it. This property sometimes makes things writable and clear. There are extra important several factors that affect the writability: *Simplicity* and *Orthogonality*.

If a language has several different constructs, these constructs may let programmer to misuse the language instead of using the language efficiently, correctly, and readably. Java language's documentation says: "Java gained its simplicity from the systematic removal of features from its predecessors, C and C++. Simplicity is Java's one of design goals. Simplicity and removal of many "features" of dubious worth from its C and C++ ancestors keep Java relatively small and reduce the programmer's burden in producing reliable applications. To this end, Java design team examined many aspects of the "modern" C and C++ languages to determine features that could be eliminated in the context of modern object-oriented programming." This explanation shows that Java has simplicity over C programming language.

Orthogonality also affects the writability. We said that in the previous paragraphs Java is more orthogonal. Fewer primitives and a consistent set of rules for coding that is orthogonality are much better than having lots of primitives. Cause of the orthogonality and simplicity features, we can say that Java is more writable than C programming language.

Reliability

If a program does its specifications under all conditions such as different operating system or different machine, it is reliable. In terms of this criteria, Java programming language is unrivaled since it has Java Virtual Machine. JVM (Java Virtual Machine) acts as a run-time engine to run Java applications. Java Virtual Machine (JVM) is known for analyzing and executing the byte code of Java. Initially, a Java program is compiled with the help of a compiler which generates the byte code of the source program. When you compile a Java code, it is generated a byte code. After that, Java Virtual Machine converts the byte code to a machine understandable code for a specific computer system. Java Virtual Machine provides code compatibility and reliability. For this reason, Java programs are machine independent. There is a Java slogan about this that suits it very well: “Write Once, Run anywhere.”. On the other hand, The C programming language, again, we must emphasize that it is low level, is not noted for its reliability. It must be recompiled. As David Evans says, “C is the language of buffer overflow (is an anomaly where a program, while writing data to a buffer, overruns the buffer's boundary and overwrites adjacent memory locations.) vulnerabilities, dangling pointers, double frees, memory leaks, undefined behavior, and race conditions (A race condition is the condition of an electronics, software, or other system where the system's substantive behavior is dependent on the sequence or timing of other uncontrollable events.).” There are other factors that affect the reliability:

- C programming language has no strict type checking while compile time and run time while Java language has powerful type-checking.
- Java supports the exception handling while C language does not support the exception handling directly.
- Also, Readability and Writability affect the reliability. Algorithms that cannot be easily implemented in a language push user to implement the algorithm in unnatural ways. These unnatural ways also tend not to work correctly under all possible conditions

At the end, we can clearly say that Java is more reliable than C programming language.

Cost

There are different factors that affect the cost. Firstly, the cost of the programmers that must learn the language. This is about learning curve, and we can say that the learning curve is related to orthogonality and simplicity. Hence, Java language, cause of the orthogonality and simplicity that we cover previous paragraphs, is cheaper in terms of learnability. The second factor is cost of writing programs. We said that Java language is easy to write so it needs less cost. After all,

high-level programming languages were originally designed to reduce the cost of coding. Also, reliability affects the cost. If a program fails in a big system, the cost would be great numbers. Maintenance is another factor that affects the cost. If a program can be improved and maintained instead of coding from the beginning, it would need less cost. Also, “Write Once, Run Everywhere” property of Java language decreases the cost since it is portable. Finally, we can summarize by saying Java is cheaper.

Efficiency

C programming language is considered as medium or low-level programming language while Java is high level. High level languages are less efficient. There are extra semantic and structural features to make the language more readable and writable in the Java language. For example, array bound checking (when you try to access an element of an array, there will be an array bound checking. This is a clear semantic difference), uninitialized variable checking, automatic garbage collector, run-time type checking, exception handling, check for null dereference. Again, as a semantic difference, Java usually works with objects and if a variable is passed to a function, it goes as a call by reference. In C, if we do not use pointers, it is sent as call by value. Also, since the strings are immutable in Java while C char arrays are mutable. String operations are expensive in terms of performance in Java. (This problem has been solved with the string builder class, even though it brings efficiency disadvantage with it.) All these features make Java heavy. C is kind of low level we even tell it how much memory we need to create a struct. Java finds it automatically. Advanced C programmers can usually tell us what the corresponding assembly instruction of a statement in the C since it is low level. It is predictable. On the other hand, you cannot do this in Java. There is so many intermediate steps. Also, C is directly compiled while Java has an extra step that is called Java Virtual Machine. C programmers can directly control hardware and memory. All this makes C much more efficient than Java.

In many ways we can compare these two languages. When we examine the sub-titles such as orthogonality, simplicity, syntax and semantic under the main headings such as readability and writability, Java seems to be ahead in many comparisons. However, it would be an extremely wrong statement to say that Java is better than C programming language. Such a judgment cannot be made for two languages, whose purpose of use and design concept are completely different. Although Java is a great language for applications that must be developed rapidly, have a maintainability focus, and have no efficient worries there is an obvious advantage of using the C programming language emerges in programs with a focus on speed, performance, and efficiency, such as system programs. The important thing here is to specify the requirements well. If you know what you need, you can choose the appropriate language for the program.

References

The Art of UNIX Programming, Chapter 4. Modularity, Orthogonality

Concepts of Programming Languages, Robert W. Sebesta, Eleventh Edition

https://www.oracle.com/java/technologies/simple-familiar.html

https://docstore.mik.ua/orelly/java-ent/jnut/ch02_14.htm

http://rust-class.org/0/pages/using-rust-for-an-undergraduate-os-course.html

https://introcs.cs.princeton.edu/java/faq/c2java.html

https://deepsource.io/glossary/procedural-programming/

https://www.whizlabs.com/blog/how-does-a-java-virtual-machine-works/#:~:text=The%20primary%20advantage%20of%20Java,from%20this%20it%20provides%20security.

https://www.geeksforgeeks.org/jvm-works-jvm-architecture/

https://techvidvan.com/tutorials/java-virtual-machine/#:~:text=The%20JVM%20converts%20the%20compiled,it%20in%20hardware%20or%20software.

https://www.quora.com/Why-is-the-C-language-considered-reliable

https://en.wikipedia.org/wiki/Race_condition

https://en.wikipedia.org/wiki/Buffer_overflow

https://www.quora.com/Are-strongly-typed-programming-languages-better-at-preventing-errors

https://www.baeldung.com/cs/orthogonality-cs-programming-languages-software-databases

https://www.quora.com/What-are-the-features-of-C-C++-and-Java-that-illustrates-orthogonality-What-are-the-features-that-illustrates-non-orthogonality

https://www.geeksforgeeks.org/c-considered-faster-languages/

https://www.quora.com/Why-is-C-considered-more-machine-efficient-than-other-high-level-languages-And-why-is-C-chosen-over-other-languages-for-writing-system-programs

https://www.quora.com/How-much-slower-is-Java-than-C-in-terms-of-performance-speed

https://www.geeksforgeeks.org/difference-between-java-and-c-language/#:~:text=C%20is%20a%20compiled%20language,JVM%20(Java%20Virtual%20Machine).

https://www.interviewbit.com/blog/difference-between-c-and-java/

https://careerkarma.com/blog/java-vs-c/

https://www.coursera.org/articles/java-vs-c

https://www.quora.com

https://www.quora.com/What-is-difference-between-Java-and-C

https://www.javatpoint.com/difference-between-c-and-java

https://www.quora.com/Are-there-any-similarities-between-C-and-Java

https://infolearners.com/java-vs-c-readability-2/

https://www.javatpoint.com/java-tokens