

## FROM GENERATION TO GENERATION

### C VS JAVA

C was invented in 1970s by Dennis Ritchie and Java was invented in 1990s by Sun Microsystems. Since Java was invented later than C, it inherited some common functionalities from C language, but their paradigms are totally different. C has a procedural paradigm which divides the program into functions, whereas Java has object oriented (OO) paradigm which divides the program into objects. Also, C has a top-down approach because of procedural paradigm and Java has bottom-up approach because of object oriented paradigm. Consequently, C cannot see the functions which are below the current line, but Java can see all of them whatever the current line is.

Furthermore, Java has access modifiers like public, private and protected, but C does not have access modifiers because these keywords are related to object-oriented paradigm which has a data hiding feature using these access modifiers. Java also has overloading, overriding, constructor, destructor features, but C does not have any of these either because of the same paradigm reason. Besides, in Java, there are objects to keep the related states inside and these states can be changed by the methods inside the objects which they belong to. Conversely, C does not have such a thing because there are no objects in C at all, hence C is more about functions than data because of its procedural paradigm.

Another important difference is that C has more control over hardware because it is a low-level language which is closer to machine language, and this gives user flexibility to control the hardware much more than Java because Java is a high-level language. This flexibility provides C to be more efficient and faster than Java because it has more control over the hardware and system resources which means users can write very optimized code for the specific machine on which they will run the code. The disadvantage of this flexibility is that the C code is likely to be unsafe because a coder which is indeed a human can make a very crucial mistake that affects the system or causes a memory leakage. On the other hand, Java does

not give that kind of flexibility to the user, and it limits its functionality for that reason. The disadvantages of Java in terms of this subject are that Java is slower than C and generally requires more memory space than C. However, Java has a big advantage in terms of safety since it does not give the user the possibility to play with low-level code. As a result, C is used in more crucial and low-level systems like embedded systems, defense industry, rockets, planes etc. Whereas, Java is used in banking, finance, e-commerce, healthcare etc.

The other important difference is the fact that C is a compiled language, but Java is first compiled, then interpreted language because of Java Virtual Machine (JVM). Java source code is first compiled to bytecode, and then this bytecode is interpreted by the Java Virtual Machine. In contrast, C source code is compiled to an executable machine code directly. Therefore, C code is directly executed on top of the OS, but Java bytecode is executed on top of the JVM which is on top of the OS. This gives Java its most loved portability feature and there is a popular saying about this such as “Compile once and run everywhere.”.

As it can be understood from the saying, once the Java code is compiled on any platform, it can be executed on any platform on top of JVM. Conversely, C code does not work like that. It is like a machine-specific code. When C source code is compiled, it is not guaranteed to run on every platform. Therefore, C has a lack of portability in contrast to Java. However, there is an advantage of C in this regard such that C code is executed more efficiently than Java because interpretation takes much more time than executing the totally compiled code.

In Java, reusability is provided by objects. Whereas it is provided by functions in C. A Java programmer can gather the related data and the methods together inside a class such that the class has one purpose specifically such as sending email, entity database operations, business operations etc. and when this purpose is needed, an instance of this class is created, and its methods are used. In C, gathering data together may be possible but it can be trickier compared to Java because the variables are either kept globally by declaring at the top or locally by declaring inside a block, but there is at least a way to provide reusability using functions. A C programmer can divide its program into reusable functions and reuse

them whenever they are needed. As a result, the reusability feature provides these languages' users a way to use their well-tested methods or functions and this feature is much more available in Java because of the data subject which is mentioned just before.

As it is mentioned before, C is closer to machine language and system resources. It is like that because it has some memory manipulating features such as pointers. Pointers are just types like integer, float, char but their purpose is keeping the memory addresses. Indeed, they have their own addresses and values like the other types of variables, but pointers keep memory addresses which is an address of another variable. This also sounds like playing with assembly because we are really getting into a deeper level of the system, we are keeping the real addresses of variables. By dereferencing the pointers, we can get the value which is kept in the address that pointer keeps. In addition, C allows users to allocate memory using malloc, calloc built-in functions. These functions get the size of memory space that user wants to allocate and allocates a memory space and returns its address, but a big problem arises from this feature such that if the user who allocated the memory space forgets to free the allocated memory, then there is a memory leakage which can cause a big trouble such that after a while there will be no memory space remaining for this program and the program will finally crash. Unfortunately, there is no precaution to prevent this kind of error in C language. In contrast, Java does not allow the programmer to deal with pointers and to allocate memory as C does. Indeed, it allocates memory when the user creates an instance using "new" keyword, but it frees this memory space after there is no reference to this instance by using a garbage collector. The garbage collector works from time to time while the program is executed and frees the allocated but unnecessary memory spaces.

Moreover, there are some differences between function calls of two languages. Java only allows call by value, whereas C allows call by reference and call by value. This is directly related to pointers, since C has pointers, user can pass a variable's address to a function and this function can manipulate the value inside this address. Therefore, call by reference is possible in C but it seems it is not possible in Java. But there is a crucial fact that Java has two types of variables as primitive type and reference type. It is not

possible for a function to change the value of a primitive type of argument, but it can change the data inside the reference type argument in Java. On the other hand, C only has primitive types.

As discussed before, Java is safer than C because of C language's memory allocation, and its flexibility that it gives to user. There is also another feature that affects the safety of languages which is called exception handling mechanism. The exception handling mechanism is used in Java, but C does not have this kind of error handling mechanism. This causes a big problem because error handling is an important topic to take an action and handle an error when it is encountered. Java allows the user to determine some actions to take when an exception is thrown somewhere in the program. This gives an ability to handle errors when they are encountered, and program can continue to its normal flow after handling this exception. On the other hand, when an error is encountered while executing a C program there is no specific mechanism to handle this error, and finally the program is going to stop execution which can be a big problem for such crucial systems like defense systems, rockets, planes, etc.

After the long discussion about the two languages' main different mechanisms, here it is worth to discuss about their syntaxes which are similar in some terms and different in another terms. C has 32 keywords, whereas Java has 50 keywords. As it is mentioned before Java is the descendant of C, therefore the two languages have common syntaxes. They both have semicolons ";" at the end of each line, curly braces "{}" for code blocks, "while", "for", "switch", "case", "default", "break", "if", "else", "else if" keywords, "+", "-", "/", "\*" operators, "//" or "/\* \*/" for comments, "int", "float", "double", "char" keywords for primitive types, and also they both have similar identifier naming conventions. Moving on to the next point after talking about their similarities, it is time to talk about their differences. The main difference is that Java has an object-oriented paradigm, so it has keywords related to classes and objects. It has "class", "abstract class", "interface" keywords for class declaration, "public", "private", "protected" keywords for access modification, "@Override" keyword to indicate a method from a base class is overridden by the child class, "abstract" keyword to indicate the method is not concrete and child classes must implement this method, "new" keyword to create an instance of a class and to get a reference of this newly created

instance. On the contrary, C does not have these keywords instead it has “struct”, “union” keywords which Java does not have. Also, C has “type\*”, “\*id” to define and use pointers, “free” keyword to deallocate the memory which is allocated before “goto” statement to jump like assembly, preprocessor directives like “#define”, “undef”, “if”, “endif”, “ifndef”, “else” in contrast to Java. Java has “try”, “catch”, “finally” keywords to handle errors using exceptions, but C does not have these kinds of keywords. Furthermore, using a library is different for these two languages. C has “#include <libraryName.h>” keyword, but Java has “import packageName”.

Now, let’s consider the readability and writability for these two languages after we discussed their syntaxes in detail. In terms of readability, both languages are good at it and have the same readability up to a level because the common syntax can be seen everywhere if one looks at two codes which are written in these two languages since they have a lot in common. However, their readabilities differentiate at some level, especially when the code is getting larger because Java has objects which are created for some purposes and these objects provide abstraction which is one of the main advantages of object-oriented paradigm. C does not have this kind of abstraction. It only has functions to provide abstraction and reusability. Moreover, Java has Javadoc which is practical for documentation as the documents can be added like writing comment lines. In conclusion, Java’s readability is better than C for the reasons discussed above. In terms of writability, we can say the same things we said about their readabilities because the same factors affect the writability as well. There is one more thing that we can add to these factors. Java’s standard library includes various objects and methods for various purposes, but C’s standard library is so smaller compared to Java’s standard library. Also today in 2023, Java is more popular than C and its community is larger than C as a result. The community affects the languages in terms of writability because the larger community the larger ready-to-use classes or functions. In summary, Java is better than C in terms of readability and writability.

The other difference which is worth to mention is that Java is highly scalable than C, and it is widely used in applications which have a lot of traffic because scaling is important for these kinds of applications to

provide availability and fast response times. Also, Java has a feature of threading whereas C does not support threading. Therefore, it is easier to create and run multiple threads in Java and this feature of Java gives programmers some kind of flexibility to write concurrent programs easily.

Another important aspect to address is their costs. The cost of a language is affected by almost all the features and the topics mentioned in this essay. One of the other factors is training a programmer for these languages. Training a programmer for C and Java has kind of same level of cost up to a level. After this level of proficiency, it is hard to master in both of the languages because if someone wants to master in C, it should have a great level of proficiency in terms of the machine it works on in low-level because as it is discussed C has features to control hardware such as pointers. Also, if someone wants to master Java, it should know the object-oriented paradigms very-well and think its design in terms of object-oriented paradigm. As a result, training a programmer cost differentiates according to the programmer's knowledge about hardware and object-oriented paradigms, so it is hard to compare these languages in terms of that factor. Another factor is maintaining the codes written in these two languages. Java's object-oriented approach makes updates and maintenance easier. Also, the use of Java's large standard library can reduce the need for writing codes for some specific functionalities and this also lowers the maintenance costs. On the other hand, maintenance and updates in C require more time and caution than Java since the programmers must always think about low-level details. The last factor to mention that affects the cost is resource consumption. Since C is a low-level language, it consumes less memory and CPU than Java. As a result, the cost of these languages differentiate according to the programmer's knowledge and the type of application.

The last important topic to mention when comparing these two languages is their applications in detail. C language is generally used in applications where efficiency does really matter even if it takes more time and caution to write error-free codes and it is also used in applications which are related to hardware. Some of the important applications where C language is used are compiler production, embedded systems, operating systems, database developments like MySQL, system applications, browsers and their

extensions' developments, IoT applications. On the other hand, Java is used in applications where efficiency does matter but not that much as the applications that C language is used and also it is used in applications where fast development is needed to meet lots of requirements by using the well-tested ready-to-use libraries which are developed by the community. Some of the important applications where Java language is used are desktop applications, android applications, enterprise applications, big data analytics, server-side technologies like Apache, JBoss, GlassFish etc.