ANALYZE ON COMPARISON OF TWO PROGRAMMING LANGUAGES: C & JAVA

In today's digital age, ability to communicate with computers is obligatory for a person who wants to participate in creating the future. As a result, people are creating programming languages to communicate with computers with ease and in a more effective way in terms of writability, readability, simplicity, orthogonality, flexibility, reliability, safety, and cost. Today, I am going to compare and contrast two programming languages C and Java which exist in top 3 in TIOBE Programming Community index considering syntax, semantics, and other things like availability, efficiency, learning curve, etc. TIOBE index shows that these languages are two of the most popular three languages in the market and that is why this essay is about comparison of C and Java.

Technically speaking, C and Java belongs to different programming paradigms which means an overall concept commonly accepted by people in a specific community. So C belongs to *Procedural* paradigm while Java belongs to *Object Oriented* paradigm. When program code is separated into the procedures, it is called the procedural paradigm. Each of these procedures follow through a single piece of work. C is compiled, low level, and purely procedural programming language which means that is the only paradigm that C belongs to. In C, there is a "main" subroutine which is the first subroutine to be called automatically when the program first executed (There is a similar situation also in Java.). Then other procedures (or functions) are called in this main subroutine and then other subroutines are called via these procedures. Program flows sequentially starting from the main subroutine in brief. This dividing code into different parts make it easier to test the code and also this process prevents repeating the same code over and over again. On the other hand, each procedures concentrate on one thing so this interfere with wholeness of the program. Another downside of this is that the usage of global variables that can be controlled by the every part of the program. This situation makes it difficult to keep track of the change in that variable. Object oriented paradigm is another approach. Actually it can not be said that Java is purely object oriented, it also has some procedural characteristics that we have discussed earlier. Java is also interpreted and high level programming language as opposed to C. This object oriented paradigm differs from procedural paradigm in a few aspects. As the name implies, object oriented programming languages focus on objects rather than pieces of code. Then what are the objects? Objects in programming world are actually related to objects in the real world. They have methods and data values to carry out some tasks. An object can be thought as package so this is one of the benefits of the object oriented paradigm. This packaging helps us to organize the code, it encapsulates both methods and data. In Java, all the code that is written belongs to a class which is template for objects. There is no data nor procedure/function outside a class. Similarly to C, program starts with main method in Java. In both C (procedural paradigm) and Java (object oriented paradigm) problem that is going to be solved is split into smaller pieces. In C these pieces are called procedures whereas in Java these pieces are called objects. Object orientation has good benefits but it makes the language a bit harder to learn so it can be said that learning curve of Java is steeper than C. In the security aspect, in C there is no proper way to hide the data but we have this feature via "public, private, protected" keywords in Java. So Java can be said to be more secure than C from the security perspective.

Before we go any further, it is good to remark that Java is coded using the C after 20 years of the development of the C. Purpose of the development of Java is then creating a more reliable, more secure, and more high performance programming language. It is mentioned before that C is a compiled language which means the code is converted into machine language. On the other side, Java is an interpreted language which means that firstly the bytecode is created from the code and then this bytecode is executed via the Java Virtual Machine (JVM). Another difference that needs to be explained further is level of these two languages. C is low level whereas Java is high level. Low level languages are closer to the hardware (machine code) than the high level languages which are closer to human language. This makes C program performs faster than Java program but speed comes with negative consequences. First consequence of this situation is ease to learn. C is more alike machine code than Java so this makes Java easier to learn and use it to write programs. Other consequence is readability of these two programming languages.

Readability, which is affected by syntax design, data types, simplicity, and orthogonality, is one of the most important criterias in language evaluation. The code has to be read in the right way in order to debug, to change, to learn how someone has solved a problem. It can be said that if a language has too many features then it is a unlikable feature for a language. Additionally, multiplicity of features is undesired property. Then orthogonality of the programming languages comes into play which means uniqueness of the language. This feature makes the language easy to learn and read because we do one thing for one thing, not too many things perform the same action. From this point of view, Java is a bit more orthogonal than C. Their readability levels are similar but the object orientation of Java makes it more readable. If some parts of the solution need to be examined, following the sequence of the program and analyzing the related subroutines are needed in C. Although in Java, the data and the methods of the objects are encapsulated in classes so every part of the solution has its own section. This makes it way more easy to read and understand the concepts of the relevant program. From another point of view, level difference of these two languages creates the divergence in the readability perspective. As it is mentioned, C is a low level language so it is closer to the hardware then Java. So this closeness is another factor that makes Java more readable than C.

Writability, which is affected by syntax design, data types, simplicity, expressivity, support for abstraction, and orthogonality, is how easy to write a program to solve a problem in the chosen programming language. Simplicity and orthogonality are needed for a programming language to be writable. Also support for abstraction is another factor that affects the writability. In this context, Java is a bit easier to write than C. These two programming languages allow expressivity which means language has comparatively convenient ways of indicating operations such as writing “flag++” is far shorter and more convenient than writing “flag = flag + 1” or “flag += 1”. On the other hand, C has pointer concept while Java has not and if you allocate memory dynamically in C, then you need to deallocate it to not to misuse the memory but in Java there is no such thing. There is a garbage collector in Java that handles with these type of situations. As a result, Java can be said to be more writable than C.

Java and C languages can also be compared with their simplicity and flexibility. Object oriented programming comes with lots of benefits but there are also downsides of this paradigm. There are so many new concepts come with object orientation such as polymorphism, inheritance, abstraction, encapsulation, class, etc. which are a bit difficult and time consuming. Therefore C can be considered as simpler from this perspective -although it has pointer, reallocate and deallocate the memory concepts- because all these fundamentals have to be learnt in order to code with Java. In other respects, C works at low level which means it is close to the hardware. Hence there are many complications that can be run into such as memory leaks, freeing the used memory, garbage collection. With Java, it is easier to work. Documentation of the already created classes make the coding with Java laborless. All in all, C is small, simpler and more straightforward in terms of concepts. On the other hand, Java is less problematic in terms of memory management and doing things like sorting the array, which can be done with one line of code. In addition to this C can be more useful when writing small programs but if project is big, Java can be easier with all of its prepared classes. The other criteria, flexibility, defines if it is easy for code to accept needed alterations and it defines if language can be used with different sorts of programs. Desired alterations need to be implemented with ease for a programming language to be flexible. You can do multiple things with using a flexible language. Memory management can be done in C such as dynamic memory access so this feature makes C language very flexible. In addition to this, regardless of the system that the program is developed, program can be executed on any operating system in Java by flexibility. Also with Java web, mobile, and desktop programming can be done. Therefore it can be said that both Java and C provides some level of flexibility.

Program behaviour may change on different platforms (with same data). This is not acceptable and has serial consequences. So any programming language must behave in the same way on different platforms to be reliable. Reliability is affected by exception handling, restricted aliasing, expressivity, type checking, syntax design, data types, simplicity, support for abstraction, and orthogonality. Memory leaks, type errors, and semantic errors are properly prevented in a reliable programming language. C has a power for accessing the memory directly via pointer concept. This is a handy way in a sense but this power comes with a downside which is accessing the memory that should not be accessed, hence this creates aliasing which is not wanted in a reliable programming language. In addition to that, C language does not have any exception handling mechanism nor error handling mechanism. Generally some functions may return -1 or null in case of any exception. On the other hand, Java has type checking, error handling mechanism by try and catch blocks. Garbage collector free up the memory by finding and deleting unused objects that is created formerly so this prevents memory issues. These all features make Java a reliable language. It can be said that Java is more reliable than C.

Java is more readable, writable, reliable, and simpler (in some aspects) than C. All these features increase the availability of Java. Although, C has a very important advantage over Java which is the cost. Cost is another criteria in language evaluation which can be analyzed in three under titles: creation, execution, and maintenance. C is closer to hardware than Java so this makes Java having very long time to debug, compile, and execute. All the benefits of Java such as garbage collection, object orientation, etc. come with a downside which is cost of time. C is far more efficient than Java if the time is concerned. If it is said that two programs, one is written in C and other one is written in Java, are equivalent in terms of semantic, then Java program executes slower due to checking and not being able to access memory dynamically. Also, a C program usually requires less hardware than Java program by reason of Java language's use of more memory. Upon the whole, Java is far more costly than C.

C is developed almost 20 years earlier than Java and Java is based on this C programming language. It can be easily said that Java language's syntax is influenced by syntax of C. Therefore if C is known, much of the Java syntax will be familiar. There are some things that exist in C but not in Java. For instance, goto statement, pointer concept (with token "*"), struct, union, enumerated type, typedef, global variables. On the other hand, Java has some keyword tokens that C does not have such as try, catch, instanceof. Java has more abstracted and distant syntax because it has some distance from the machine code. Nevertheless, Java is based on C so there are some sharp similarities between syntax of Java and C languages. Assignment, blocks, conditions, loops, decleration of a variable and function are same in both Java and C. There are also some differences between these two languages in terms of syntax. For instance, when an integer array is needed to be declared, in C it can be written directly as "int array[10]" (where 10 is the size), but in Java “new” keyword is needed: “int[] array = new int[10]”. Other than syntax, semantic is very important thing which means the meaning of the programming language itself. There is a concept in Java similar to pointer concept in C, which can be helpful to understand syntax and semantic differences of C and Java better, called reference. Actually there is no reference concept in C (there is simulated call by reference in a call for a function) but pointer concept can corresponds to reference in Java. Semantics of pointer in C are similar to reference in Java but these two are not similar in terms of syntax. In addition to this, C does not have a boolean data type while Java has it. There are primitive and non-primitive data types in Java. C has some of the primitive data types such as int, double, char and some of the non-primitive data types such as arrays, which is derived data type in C. In other respects, C does not have any boolean nor String as a data type whereas Java has.

In conclusion, C and Java are widely used in today’s digital age as it can be seen from the TIOBE index. They both have some advantages over the other one. Therefore it would not be true to say that Java is better nor C is better. Specifying the needs of the program which is wanted to be designed and then choosing the proper language according to its advantages is the fundamental point.