A Comparative Analysis of C and Java Languages

We are living in a digital age. Many problem that can be converted to digits can be solved by the computers. Computers have led to a new revolution for civilization. Today's automobiles controlled by computer which improve fuel efficiency via engine controls and increase safety such as the air bags systems. Another example could be Human Genome Project which requires computers mapping and analysing process of human DNA sequences. Clearly, advances in computer now affect every aspect of our society and enables us to solve the today's problems. But how computers do all the magical things? First, computers just execute the programs which contains set of instructions that tell computer how to perform a task. The first programmers communicated to computers with the machine language which is the binary representation of machine instructions, and it is the only language computer can understand. For example, the bits 1000110010100000 tell computer to add two numbers. But this was so tedious that the pioneers quickly invented new notations and languages that were closer to the way human think. The assembly language is the first of them. It is a symbolic representation of machine instructions, and these symbolic instructions are translated into the binary version by a program called assembler. For example, the programmer writes add A, B and the assembler would convert this notation into 1000110010100000. Although it is a huge improvement, assembly language is so basic that each instruction of task must be written, and this requires programmer to think like computer. After the recognition of there can be program which can translate a more powerful language into computer instructions one of the greatest breakthroughs in the early days of computing. Using these converter programs such as compilers and interpreters, new programming languages are created. Today there exist thousands of programming languages and each provide different abilities for specific uses. The ones titled as high-level programming language are closer to natural language and contains powerful abstractions. This abstraction enables programmers to create programs focusing on the business level without thinking the underlying low-level details. Design of the language determines the characteristics of the language that make it good or bad for a specific task. Many of the high-level languages are designed to enhance the syntax and semantics of C programming language. So, it is useful to understand the evolution of C language before analysing our target language Java.

C is a general-purpose computer programming language. It was created in the 1970s by Dennis Ritchie. It is an imperative procedural language with support of structured programming and recursion. It was designed to be compiled to provide low-level access to memory and language constructs. C is commonly used on system programming as well as application programming. It was developed at Bell Labs by Dennis Ritchie to construct utilities running on Unix. The kernel of the Unix operating system reimplemented with C. During the 1980s, C gradually gained popularity and it has become one of the most widely used programming languages. Java is a high-level class-based, object-oriented programming language. It was developed by James Gosling at Sun Microsystems. It was released in May 1995 as a core component of Sun Microsystems. It's a general-purpose programming language which is designed to have as few implementation dependencies as possible. Java applications are compiled to bytecode that run on any Java virtual machine (JVM). Java offers write once, run anywhere (WORA) functionality. Regardless of the underlying computer architecture, compiled Java code can run all platforms without recompiling process by the support of JVM. There are several fundamental criteria's that is used when evaluating a language. These are readability, writability, reliability and cost. Each of these criteria is determined and influenced by a certain number of language qualities such as simplicity, orthogonality, data types, syntax design, support for abstraction, expressivity, type checking, exception handling and restricted aliasing.

For simplicity both C and Java uses multiplicity of features. For example, incrementation can be done in three ways with simple assignment statement (i = i + 1), postfix increment operator (i++) or prefix increment operator (++i). Loops would be another example for multiplicity of features. C has three different loop structures which are for, while, do-while. Java has also for-each loop in addition to the other three types of loops in C. Multiplicity of features is not good for readability because it makes reader to know all the possible operation to achieve the same result (there can be slight difference between them but indeed they all achieve the same result). But since C prioritize fast execution, it is not a big deal for it to add extra control structures to increase the performance a little bit more. Another example of simplicity problem is operator overloading. Since it can increase the readability and writability of the language, it is a very useful feature in a language. However, if the overloading does

not make sense, it can cause confusion. Although Java is highly influenced C++, the designers of the languages prefer to include operator overloading. On the other hand, excessive simplicity can cause a problem too, but both languages simple enough that does not cause confusion. Java is better in terms of readability and writability characteristics due to its OOP support and abstraction power. On the other hand, C is simpler because there are smaller number of constructs that C programmers should learn. In terms of expressivity both Java and C becomes weak when they are compared to other high-level languages such as Python.

An orthogonal programming language would be where each of its features have minimal or no side effects, so they can be used without thinking about how that usage will affect other features. Enhanced orthogonality in a programming language increase both readability and writability and makes more options for their users. On the other hand, lack of orthogonality leads to exceptions in the behaviour of the language. C is not considered orthogonal and there are a few examples that can prove that. For example, arrays cannot be returned from a function directly, however it can be returned in structures. Void can be used as a type in a structure, but a void type of variable cannot be declared in a function. As an example, from the Java could be applying public and static to a method. They do not interfere with each other, so these two keywords are orthogonal. But it's hard to say C or Java is completely orthogonal or not.

Data types and data structures are also important for both readability and writability. Java has a primitive Boolean type named boolean, used mainly for the control expressions of its control statements. Unlike C which has no primitive Boolean type, arithmetic expressions cannot be used for control expressions. As an example, think an integer variable named is_end which is used as flag in a loop. A statement like is_end = 0; is less readable than is_end = false; which can be done in Java with boolean data type. Java provide packages that contains useful data structures such as list (ArrayList, LinkedList), queue, set (TreeSet, HashSet, ConcurrentSkiplistSet), map (TreeMap, HashMap). On the other hand, there is no standard library which defined these data structures in C. A C programmer had to implement his/her library, since there is no standard library or package like STL in C++ or java.util in Java. Then another programmer who uses this library needs some documentation to understand how

to use them. In terms of built-in data structures C becomes weak but this also make it light by avoiding too much code comes from the libraries

The syntax of a language highly influences both readability and writability. In a well-designed language, semantic follows syntax. Although the expressions, assignment statements and control statements of C and Java nearly identical, there are slight differences. Firstly, the pointers of C can be manipulated in variety of ways which supports highly flexible addressing of data, but they are not included in Java because of the potential reliability problems. Java classes and arrays are reference types, and references to objects or arrays are like pointers of C. However, there is no way to increment or decrement a reference like pointers. There is no address-of operator &, dereference operator *, or sizeof operator. C allows function pointers which can store the address of a function and pass this function to other functions as an argument. On the other hand, Java does not have such a feature. C supports pre-processors that is used automatically by C compiler to transform the program before the compilation. C macro pre-processor enables the inclusion of header files, macro expansions and conditional compilation and other things. On the other hand, Java does not support pre-processors. It just uses import directive to include the other packages. Constant definitions are replaced with static final fields in Java. Java defines a very clean namespace. Since Java has cross-platform portability, conditional compilation is not actually required. Packages contain classes, classes contain fields and methods, and methods contain local variables. However, there are no global functions and variables as in C. Every method and fields should be part of a class. C requires local variable declarations to be made at the beginning of a function or block, but Java allows them anywhere in a method or block. The Java compiler allows methods to be invoked before they are defined. This eliminates the need to declare functions in a header file before defining them in a program, as it is done in C. Use of goto is a poor programming practice except in certain situations and Java doesn't support goto statements. Both C and Java uses semicolon at the end of the statement and pair brackets for scoping a block of code. Java does not support the typedef keyword used in C to define aliases for the names. Java does not support C struct and union types. Instead, it has class type that can be thought of as an enhanced struct which can contain functions in addition to data. After the definition of C struct and union, semicolon had to be

placed but in Java it's not required for class definition. C has the arrow operator (->) to use with pointer when accessing the elements of struct or union. In Java public methods and fields can be access with dot operator (.). C does not have the keywords public and private which are the indispensable for Java which adopts OOP principles. C does not support virtual functions but in Java all the methods are virtual by default. For array element lookup both language uses square brackets, but the declaration of array is a little bit different. For example, an array of int is declared with int[] arr = …; in C, but int arr[] = …; in Java. In C a comment starts with a slash asterisk /* and ends with an asterisk slash */. On the other hand, Java has double slash for single line command in addition to C style comment.

Memory management is also an important aspect that is considered. C has user-based memory management, but Java does automatically and does not allow programmer to access memory. With the power of pointers, memory can be manipulated with pointer arithmetic and the standard library functions malloc and free. In Java, memory allocation can be done by new keyword, and memory deallocation is implicitly handled with garbage collection mechanism. This frees the programmer from needing to delete objects explicitly when they are no longer needed. So, it is easier to focus on developing the business logic in Java. In addition to automatic memory management, threading is also supported in Java, whereas C does not. This is also a big advantage in Java because it enables to do simultaneous work.

Reliability is another language evaluation criteria. Java has exception handling mechanism which is not included C. Both languages have static type checking and programs that are not well-typed are rejected with an explanation of which rules were broken. As an example, Java demands that all references to array elements be checked to ensure that the indices are in their legal ranges. This process highly effects the execution of Java programs that contain large number of references to array elements. C does not require index range checking, so C programs execute faster than semantically equivalent Java programs. This makes Java more reliable but much slower compared to C. Two language provides aliasing, and it is done with pointers of C and references of Java.

In conclusion, C and Java's main key difference is the programming paradigms that they are designed. C is a procedural programming language which means series of well-structured steps and

procedures specified with in programming context to compose a program. Java is an object-oriented programming language which is based upon objects (having both data and methods) that aims to incorporate the advantages of modularity and reusability. Since C does not support the OOP's concepts such as inheritance, and polymorphism, necessary level of reliability was not provided in it. On the other hand, fundamental goal of Java was providing a greater simplicity and reliability than C/C++. In addition to provided reliability, Java has rich libraries and features with OOP support. That's why it would be better language for one has to relate things according to real world. However, when high performance is needed, obviously C is the winner.

Sources:

- *C (programming language)* (2022) *Wikipedia*. Wikimedia Foundation. Available at: https://en.wikipedia.org/wiki/C_(programming_language) (Accessed: October 12, 2022).
- *C vs java: Top 13 most vital comparison you should know* (2022) *EDUCBA*. Available at: https://www.educba.com/c-vs-java/ (Accessed: October 12, 2022).
- *Difference between C and java* (2022) *InterviewBit*. Available at: https://www.interviewbit.com/blog/difference-between-c-and-java/ (Accessed: October 12, 2022).
- *Difference between java and C language* (2022) *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/difference-between-java-and-c-language/#:~:text=C%20is%20a%20compiled%20language,JVM%20(Java%20Virtual%20Machine). (Accessed: October 12, 2022).
- *Java (programming language)* (2022) *Wikipedia*. Wikimedia Foundation. Available at: https://en.wikipedia.org/wiki/Java_(programming_language) (Accessed: October 12, 2022).