**Java & C++ Analysis**

Marques Traylor

Colorado State University Global

Course Code: CSC470

Oct 4, 2024

**Java & C++ Analysis**

Java and C++ are both high level programming languages. They are used to create a variety of different types of applications. This may range from desktop applications to embedded systems applications. After implementing concurrency in both languages, I believe Java is less prone to security vulnerabilities than C++. This is because Java contains automatic garbage collection which handles memory cleanup. However, both languages have some similarities, but also differences.

**Similarities**

Java and C++ both have the ability to create robust object oriented programming (OOP) applications. This means that they are able to utilize classes to mimic real world objects and encapsulate data which makes it easier to develop applications for everyday use. It is simple to switch between the languages due to their similarities in syntax. For example, both use curly braces and semicolons. They also share the same primitive data types like string, int, float, etc… Finally, they both allow for abstraction which means that the developer is capable of hiding implementation details in order to focus on interfaces.

**Differences**

The differences between both languages can be the decisive factor when choosing which one to develop in. Java only supports OOP while C++ is capable of procedural programming and OOP. Java is platform independent while C++ is platform dependent this means that a developer may create Java applications on any platform while C++ requires a specific platform in order to compile and run the code. Error handling in Java is implicit, while it is explicit in C++. Java does not contain as many features that C++ does, like multiple inheritance for example, however, it does have its own features that C++ does not have, like interfaces and automatic memory management. A big difference is that Java contains automatic memory management which helps prevent memory vulnerabilities while C++ requires the developer to manage memory throughout the application which can cause issues if it is not done properly. Ultimately, Java prioritizes ease of use, portability, and platform independence, while C++ prioritizes performance, control, and flexibility.

**Security**

Since Java contains automatic memory management, it is more difficult for an attacker to exploit memory vulnerabilities. This is an issue with C++ because it contains manual memory management which can lead to several issues like memory leaks, direct memory access, and memory overflows. “C++ has what experts refer to as “memory unsafety,” which refers to vulnerabilities in the memory codes” (Java vs. C++ Comparison: What Are the Differences in These Programming Languages?, 2024). In both concurrency applications, the safety provided by Java is greater than that of C++. This is because the Java application does not utilize pointers and is capable of automatically cleaning up the memory that the threads have used while executing. In the C++ application, it utilizes a pointer that is shared between both threads which can result in an exploit through memory vulnerabilities.

**Conclusion**

Java and C++ are popular languages used in many applications. Java is capable of being used on any platform and provides automatic memory management. C++ is platform dependent, however, it is able to create powerful large scale applications and offers manual memory management. Java is a more secure language than C++ due to the fact that it does not allow the developer direct access to memory and provides garbage collection which removes unused memory during and after program execution. This can be seen in the concurrent applications written in both languages where the Java one executes the threads with no vulnerabilities while the C++ application utilizes a pointer that is accessed by both threads, but can be vulnerable to memory issues.

**References**

Java vs. C++ Comparison: What Are the Differences in These Programming Languages? (n.d.). Coursera. <https://www.coursera.org/articles/java-vs-c>