Vincent Fazio Program Assignment Two Paper

C++ and Java are two powerful languages that seem similar in nature but are actually vastly different. Despite both deriving from C the two languages handle the way they pass data between variables and functions very differently. The two main techniques for passing data around is known as passing by value and passing by reference. C++ being a more verbose language than Java lets the developer handle what technique to use while Java being a stricter language forces implication on the developer.

Passing by value and passing by reference are the two most known techniques for passing data around. Passing by value is the copying of data, where changes to the copied data are not reflected to the original value. Passing by reference is the aliasing of data, where changes to the aliased memory value is reflected in the original value.

C++ being the verbose and more flexible language handles both pass by value and pass by reference as the definition states. The standard state in which data is passes data around is pass by value. However, C++ gives the developer the choice to use either pass by value or pass by reference. The language does this because the developer is able to explicitly pass data by value or by reference. This means it is up to the developer to decide what to choose, and any mistakes or errors that comes from the choice are implicitly implied to be handled by the developer. The C++ compiler simply checks to see if the developer’s choice is implemented correctly, it does not check to see if certain references are creating undesirable outcomes. This gives the developer more freedom but certainly hurts the developer since these errors won’t be seen by the compilers.

Java is known as C++ with less, a more friendly C++. One reason for this is the way Java handles passing data. Java states a simple rule when confronted with how it does pass of data. Java declares that all passing of data, both objects and primitive are to be ‘all passed by value’. Simply meaning that there is no passing by reference in Java. While this is true to some extent, with some digging and testing you find out that it isn’t entirely true to its ‘simple rule’. For primitive data types Java does pass by value, but what’s interesting is what you get when you pass objects. Java has no means of explicitly differentiating between the two techniques, pass by reference and pass by value. This removal of functionality is why Java states its data is all passed by value. However, when a value is associated with an object, the object is actually a pointer, meaning a call by reference, to the object in memory. This is known as object reference in Java. It isn’t a complete call by reference though, C++ uses aliasing within its call by reference which is why the data value changes. In Java the pointer is passed into the function, which is used, but when the function is exited the pointer is lost, meaning that the new value is unable to change the pointer value. This allows for a safer and less error ridden programming environment but hinders the developer’s ability while keeping a door shut on how it runs in the back end.

C++ and Java both pass data differently with their own definitions of the two techniques passing by value and passing by reference. C++ allows more freedom within its language giving the developer the ability to develop efficient programs through strenuous debugging and testing. While Java restricts the developer’s ability for their own coding, creating an environment that is more rigid then C++ but easier to work with, resulting in less errors at the expense of freedom. It’s hard to say whether I think C++ or Java as the better language, or that Java is a step up from C++. I would say it depends on the task given, or how the question is posed. If it’s about flexible and efficient you would say C++ is the better language. If it’s about how Java offers a better and more simplistic environment with less error prone code, then Java.

From what I understand I would say that Java is a step up from C++. Despite it being less powerful, C++ predates Java, and Java was created from the image of C++. So, as a programmer I see it as the developers who made C++ took the opportunity to make it better despite creating a language that isn’t as powerful or free as C++.