3. Would you choose Fortran 95 for your team of scientists or would you use C++ or Java? Explain the pros and cons.

During my outline for the essay from last week I learned a bit about C++ and Java. During the second assignment for this week we all had to learn about Fortran 95. The most experience I have in coding is in Java, so I may be partial to it, but that is my language of choice for my team of computer scientists.

Let’s first discuss why we are not using C++. In the first paragraph of our Fortran chapter in the textbook we learn that built in code threads make programming faster and easier, and it is one of the reasons why Fortran was built. C++ does not support this. C++ has a manual memory management system, which if done improperly could make the runtime incredibly long. I think that time and energy spent on memory management is time and energy wasted. C++ is often said to be a difficult language to understand, which is another reason why we are not choosing it. C++ is also platform dependent, which is a huge disadvantage if our team wants to make a code that is available cross platform.

C++ does have some value to it of course, it is considered a powerful computing language. This language is also very easy to scale. C++ is generally considered to be a faster language in comparison to Java.

Now some might think that Fortran 95 is much older than Java, however Fortran 95 was released a year AFTER Java was. The reason we are choosing Java over Fortran is because of its popularity and ability to be used across multiple platforms. Doing a simple search for Fortran, we can easily see that this language is no longer popular. I can then assume that my team probably is not educated enough about the language and most likely lacks experience in it. Of course we can assume the older generation of programmers would be familiar with this language, but all of the software engineering students within that last 10 to 20 years would most likely not be. Fortran has an extensive library of default values and rules about definitions that are rather lax. This makes finding and recognizing errors very difficult. While Java requires a declaration of variables beforehand, making the room for error much smaller. Before the 2000s Fortran also had a similar memory model to C, which required manual memory allocation. It is also commonly said that Fortrans points often cause errors.

Fortran is generally considered to be a faster language in comparison to Java. Fortran also supports parallel programs in their 95 release. Fortran is of course considered to be the origin of computer programming. Its use for scientific computing can be said to be unparalleled to any other computer language of its time and probably even now. It is still used in supercomputers today.

Fortran is primarily used for scientific computations. C++ is best suited for developing large software management systems, and libraries. But Java is best used for internet application software, and internet programs. All of these languages do support object oriented programming.

The reason I am choosing Java is because it is easy to learn, very popular and most computer scientists do know how to program in Java. So I can easily assume that my team would have the experience and education necessary to use Java and be successful in it. Java most easily goes across the most platforms out of these languages with the least errors. Java is easy to debug and errors are easy to find and fix. Java supports many of the scientific computations that Fortran would be used for. This language supports inheritance, polymorphism, abstraction and encapsulation. Java also supports loops and conditional statements just like Fortran and C++. All of these languages have a lot of similarities like loops, condition statements and more. But I think Java would be the most applicable to software development today. Of course the question does not mention what the developer team is solving or doing. But we can assume that Java could get it done and apply it to websites and applications the MOST easily for users. Personally I think the language choice is specific to the problem the team has to solve or the job they have to get done. C++ and Fortran are great languages and can support a lot of things and do them quickly. Java is best suited for client-server applications and is used by large businesses around the world. You can write it once and run it anywhere. It is also used to program Android phones. The popularity, common usage, ease of use, and ability to be supported on any system is why Java would be the best choice for a team of programmers. Whatever the problem is, it can be solved in Java, used anywhere, scaled to be huge and users would have an easy time with it.