# Program #4 Review

***Written by Alexander DuPree***

## Java Discussion

Program 4/5 was required to be written entirely in Java. As such, switching from C++ to Java was difficult, but manageable. Java's structure and syntax is much the same as C++, however, there are certain key differences that make writting a program in Java difficult at first, but in the long run much easier. The largest setback I had coming into Java was understanding how Java handles memory and references to objects. First, as a C++ programmer I have developed the habits of being strictly in charge of memory allocation/deallocation. It was always important to understand the lifetime of your objects and when they would be destroyed so that there are no memory leaks. This is not the case in Java, as Java provides a garbage collector that handles the destruction of all objects in the system. This made it weird, at first, to write code in Java as any given file could have a number of "new" statements that were never accompanied by a "delete" statement. I found myself questioning if my code was "safe" because I never could know when an object was/if ever correctly deallocated. Once I came to terms with the invisible prescence of the garbage collector, I found that classes were much easier to write and manage as no object ever managed their own memory.

Another obstacle for writing code in Java was understanding how references worked in terms of writing a recursive algorithm. In C++ it was simple to write a recursive function, as the parameters for the method could point to/directly refence a parent's child pointer, and changing/reassigning that reference would change the Parents child pointer as well. In Java, the parameters for a function are only LOCAL references to an object, and changing/reassigning that reference has no effect on the parent or even the previous stack frame. This meant that for a change to persist I had to return the result of the reassignment to the previous stack frame.

After a few hours frustrated about how Java did things compared to C++ I found that it was my attitude that held me back from writing a successful program. I found that I needed to adjust my mentality from that of a C++ programmer to one that was open to the new constraints and features of the Java language. After this attitude adjustment, it was easier and much more simpler to write code that took advantage of the many different Java constructs.

## IDE Discussion

Java wasn't the only change required for program 4/5, we were also allowed the use of IDE's to implement our programs. VIM will always be my preferred development environment when working in a terminal, however, the power, accessibility, and quality of life improvements when working with an IDE made writing/testing the program much easier. I used the intelliJ IDEA for my development environment and after spending some time customizing and learning the features of the IDE I can say it will be hard to go back to the bare bones text editing of VIM. With the intelliJ IDE I could spin up a suite of unit tests within seconds, run those tests and debug them with a few clicks, and even perform a lengthy refactor through a guided contextual menu. The combined ability of all these features along with autocomplete, file navigation, syntax highlighting, and much more made writing code an enjoyable and simple process.