Name: Travis Stop

Class: Programming Languages Design and Implementation (CSC4510)

Professor: Dr. Michael Oudshoorn

Date: 11/8/2020

Parameter Passing Techniques in C++ and Java

C++ and Java are programming languages that are related to each other such that Java has evolved from C++. As one would expect from this relationship, many features of C++ are present in Java and vice versa. The purpose of this essay is to describe the techniques used to pass parameters to functions in C++ and to methods in Java, as well as evaluate these techniques in relation to one another. We will also determine whether or not the parameter passing techniques in Java are an improvement from the parameter passing techniques present in C++.

First it will be beneficial to briefly describe some terminology. Keep in mind that I will use the word “function” to refer to functions in C++ as well as methods in Java from here on out, unless I feel the need to make a distinction. There are two kinds of parameters: actual parameters and formal parameters. Actual parameters are the variables that are passed into the function, and formal parameters are the variables that represent the parameters within the function. There are several different techniques that are used to pass parameters into functions, and I will describe the two I found to be the most relevant here. The first technique is commonly referred to as “Pass By Value”. For this technique, the value of the actual parameter is copied to the formal parameter. Changing the value of the formal parameter within the function has no effect on the value of the actual parameter in this case. The second technique is commonly referred to as “Pass By Reference”. In this technique, the address of the actual parameter is passed rather than the value. In this case, changing the value of the formal parameter within the function also changes the value of the actual parameter.

Now I will begin discussing the techniques used in Java as well as C++. For both C++ and Java, if the parameter is a primitive data type the technique used to pass the parameter is Pass By Value. For both C++ and Java, the technique used to pass the parameter is Pass By Reference if the parameter is an array. C++ and Java are different in how class objects are passed to functions. In C++, class objects seem to be passed by value however in Java class objects appear to be passed by reference. One thing that is extremely important to note is that C++ gives the programmer the ability to explicitly pass any variable, for example a variable that contains a primitive data type, by reference via the use of pointers. Java does not give the programmer the ability to explicitly create pointers or decide whether or not to pass a parameter by reference or by value. This is one of the most profound differences I found between Java and C++.

Now I will discuss whether or not the adjustments made between C++ and Java are improvements. The major difference as I previously stated is the ability to use pointers. The use of Pointers in C++ can pose a security risk and is also error prone. However, the ability to use pointers also gives the programmer the ability to manage memory on their own and perhaps manage it more effectively. It could be argued however that the way Java forces a programmer to program is the optimal way because many errors in C/C++ result from the use of pointers.