**The Difference Between Java and C**

Java and C are two distinct programming languages that exemplify two different fundamental programming concepts of the four main types of programming paradigms used as models of computation by most, if not all, programming languages; object-oriented, functional, imperative and declarative. To the untrained eye, comparing the syntax of these two programming languages may lead to an assumption that the two languages are similar since both have the same arithmetic operators, primitive data types, similar loops and each statement terminates with semicolons. (Maassen)

Java and C are two very different programming languages and the difference lie in the fact that Java is considered an object-oriented programming language and C is considered a functional programming language. A data structure in C can comprise of identifiers of various data types that may or may not refer to locations in memory of the computer (has some kind of value or null). Members of an instance of a data structure obtain assignment of values to its members by manipulation from a collection of procedures. Generally speaking, C programs are structured in a manner in which computation is achieved through the use of various functions that can take input values and return values. These values can then be used to make a decision on the next computation to make. (Maassen) The control flow of a program is based around the “main” function that is in charge of passing in and obtaining values from other functions.

In Java, data structures (called objects) are more than just a group of data because they also have behavior associated with them. The behavior (method) of an object can perform operations based on values of its data and interact with other objects of the same type or different type by passing data between each other. (Dev) In object-oriented programming, programmers write the definitions for classes, which are “templates” for making instances of the objects derived from that class. When a class definition needs to be written for an object that has more methods and fields than a class already written and available but otherwise the same, inheritance allows for the creation of a subclass, and the programmer is able to reuse code already written. The programmer simply needs extend the superclass and write definitions for the methods and fields that are not present in the superclass it was derived from. If the subclass needs to implement a method it inherited from its superclass differently, the subclass can redefine the method with the same name and parameters. This type of redefinition called “Method Overriding”.

Classes can contain more than one definition for a method with the same name, as long as the number of arguments or the type of argument differs between each definition (signature). This type of redefinition is known as “Method Overloading” and allows for an object to behave accordingly based on the type or number parameters passed in as parameters to its method. Inheritance, method overriding, and method overloading allows polymorphism in Java. Even though subclass objects inherit from a superclass, which itself may have inherited from a super super class, polymorphisms allows derived objects to take on many different forms which are not necessarily identical to the superclass it was derived from. (What is method overloading and overriding in Java?)

Another difference between these two languages lies in how the programs are compiled from source code. In C, to compile the source code, the programmer runs *“gcc <sourcefile.c>”* from a terminal (or command prompt). This command calls the C-compiler, which compiles the source code into machine language that can be loaded and executed. Source code that is compiled into machine language on one machine cannot be executed on another machine; therefore, recompilation is necessary for different architectures and portability is not possible for compiled code in C. In Java, calling *“javac <javafile.java>”* from a terminal causes the Java compiler to generate what is known as “Bytecode”, which is interpreted by the Java Virtual Machine (JVM) for the program to execute. The JVM is available for many different platforms, and because of that, the same bytecode is capable of being executed on two different machines. This is why Java is often described as “write once, run anywhere”. (Worcestor Polytechnic Institute)

Dynamic memory allocation is a subject discussed often in C programming. When a C program takes input where the length/size in unpredictable, if the input is larger than the amount of space allocated, data is lost. Also, if the input ends up being much smaller than the allocated size in memory, memory is wasted and the program ends up being not as efficient as it could be. The programmer must handle the task of managing the memory by freeing unused memory, allocating, and reallocating memory when needed. In Java, memory manage occurs automatically in a process called “Garbage Collection”. When reference variables no longer point to objects on the heap of memory (meaning it now points to null), the JVM will remove the object previously pointed to by the reference variable from the memory permanently (Java Tutorial 14 - Garbage Collection).

Exceptions in Java are can either caught by the compiler (checked exceptions) or identified after compilation at runtime (unchecked exceptions). Java provides syntax to explicitly declaring blocks of code where an exception may occur. For checked exceptions, the compiler forces the programmer to declare and handle the exception, while unchecked exception may go unnoticed until the program is compiled. (Ghosh) C does not have explicit support for handling errors. Instead, the programmer is expected to structure the code in such a way that the error does not occur in the first place. There are keywords specific to handling errors in the code.(WikiBooks)

Overall, I believe that Java is by far a superior programming language because common tasks, such as converting integers to strings, are accomplished through methods that are already defined in Java. Thus programmers have the ability to compose programs with more efficiency, without wasting time writing code for common tasks.

# Work Cited

WikiBooks. "C-Programming/Error Handling." 12 04 2013.

What is method overloading and overriding in Java? 12 April 2013 <javarevisited.blogspot.sg/2011/12/method-overloading-vs-method-overriding.html>.

What is Polymorphism in Java. 12 April 2013 <http://javarevisited.blogspot.com/2011/08/what-is-polymorphism-in-java-example.html>.

Worcestor Polytechnic Institute. "Department of Computer Science." Worcestor Polytechnic Institute. 12 04 2013 <web.cs.wpi.edu/~cs2303/c10/...C10/Week0\_Differences.ppt>.

Brian Harvey, Matthew Wright. Simply Scheme: Introduction to Computer Science. Cambridge: MIT Press: Cambridge Massachusetts, 1999.

Dev, Rishabh. 10 Major Differences Between C And JAVA. 8 October 2010. 12 April 2013 <http://www.durofy.com/programming/10-major-differences-between-c-and-java/>.

Ghosh, Arup. COP3330 (Spring 2013) - Exception Handling. Orlando, 01 2013.

Java Tutorial 14 - Garbage Collection. 12 April 2013 <http://www.javatutorialhub.com/java-garbage-collection.html>.

Maassen, J. "C For Java Programmers." University of Texas in Austin (Computer Science). 12 04 2013 <http://www.cs.utexas.edu/~ans/classes/cs439/dictaat.pdf>.