Homework 1

COMP 3220

Due: August 28, Thursday by 11:59AM

Please submit as a PDF or WORD document using Canvas

(1. 30pts) Write a detailed evaluation of a programming language you know, using and covering the criteria discussed in class (and in Table 1.1 of Chapter 1).

- In my evaluation, I will review Java.

1. By nature, Java’s readability and learning curve is not hard due to its relative simplicity. Its constructs are generally easier to grasp, and some consider it an alternative to C++. Java has feature multiplicity and operator overloading which create potential problems; however, when used correctly and sensibly, these should not affect readability as much. Java is not so simple as to hinder its readability with many statements to do one thing, like in Assembly.
2. Java is by no means absolutely orthogonal, but in the regard to most primitives and combinations, there is a lot of orthogonally when combining objects.
3. Java is strongly typed. Just about every usable construct can be given a desired data type, thus significantly aiding in readability.
4. Java helps in readability with its syntactical design which may flow similarly to sentence structure. In most regards, the curly brackets help in determining the scope and ends of statements, instead of reserved, special words. In Java, special words are reserved and cannot be used in naming. Java has multiple contextual meanings for reserved words.
5. Java supports abstraction, both process and data abstraction. An example of process abstraction: a subprogram can be written to implement a sort algorithm that is required several times in the main program. An example of data abstraction would be: binary trees can be implemented by using an abstraction of a tree node in the form of a simple class with two pointers (or references) and an integer.
6. Java is expressive and is generally not far from pseudocode. In The inclusion of the ‘for’ statement in Java makes writing counting loops easier than with the use of while, which is also possible. All of these increase the writability of a language.
7. Java is very efficient with type checking. The design of Java requires checks of the types of nearly all variables and expressions at compile time. This virtually eliminates type errors at run time in Java programs.
8. Java happens to be a language that does support exception handling. Exception handling is useful in catching exceptions and handling them with corrective measures. Exception handling is also useful in testing code.
9. Java is restricted in aliasing and from what research I have put forth, I believe the Java-coding community wants better support.

(2. 15pts) Name and explain another criterion by which languages can be judged (in addition to those discussed in the chapter).

- Portability could be a good criterion to judge a language by. Many languages have too many implementation dependencies. Java is more portable than C++ but is not as fast as C++, so there is a trade-off to consider.

(3. 10pts) What are the advantages in implementing a language with a pure interpreter?

- Pure interpretation has the advantage of allowing easy implementation of many source-level debugging operations, because all run-time error messages can refer to source-level units. For example, if an array index is found to be out of range, the error message can easily indicate the source line and the name of the array. Other advantages include dynamic typing, reflection, platform independence, and dynamic scoping.

(4. 10pts) Some programming languages -- for example, Pascal -- have used semicolon to separate statements, while Java uses it to terminate statements. Which of these, in your opinion, is most natural and least likely to result in syntax errors? Support your answer.

- I think the semicolon should be used like it is in Java, for this is not only a Java implementation but for many languages which stem from C. It would be considered more natural for programmers familiar with C and would result in less syntax errors. Paired with the curly brackets, the termination of statements with a semicolon makes viewing scopes and blocks of logic more readable in my opinion and from what I deem to be normal conventions.

(5. 15pts) Describe some design trade-offs between efficiency and safety in some language you know.

- Java is less efficient in the regards to garbage collection, but the tradeoff is that safety is added in and less time is spent on resource management.

(6. 10pts) Do you think language design by committee is a good idea? Support your opinion.

- Yes, I think it is a good idea. If we see the development of the most popular languages like C, C++, Java, etc., it has its seed in the minds of a single or a small group of people. As the language catches the imagination of a larger audience people try to add their own features or extensions to the language and leads to the development of various dialects which may be incompatible with each other. Therefore a committee has to intervene to define the standards that all compiler and other tool vendors have to adhere to.

(7. 10pts) Explain two reasons why pure interpretation is an acceptable implementation method for several recent scripting languages.

- If it is a pure interpreter (like most shells), the interpreter reads each line/instruction, analyzes it, and executes it. If it is a pure compiler, the program (or a part of it) is read in, analyzed, and then the machine language is written out to run later. Interpretive languages allow for better platform independence because they lack the need for compilation, and they also have a smaller run time size.