**Assignment 1**

Problem set questions:

2) What are some features of specific programming languages you know whose rationales are a mystery to you?

While I was reading chapter one, I noticed that there are a lot of features in programming languages which I do not know at all, especially since I don’t have many experience as a programmer. One feature that caught my attention and that we were talking about it on class was the use of the loops.

For example: “for” and “while” loops are both present in many languages. We all know that one can easily be simulated using the other. So what is the use of both for and while loops. I do understand that is to make it more simple, but at the same time I wonder why only for and while loops, we could have use more loops or perhaps use less loops. That is something that only could be explained by the person that created the programming language

At the end I think that Knowledge of the programming language characteristics can help all programmers, the knowledge of the characteristics of programming languages make you a better one. The programmers if know the implementation details of some construct, they can understand for what tasks, this particular construct will be more efficient.

6) What common programming language statement, in your opinion, is most detrimental to readability?

I am no expert in programming languages and in most of my codes I have never encounter something that is really hard to read, but as I was researching I noticed that there is an statement called “goto”.

I believe that this one could one of the most detrimental to program readability,

It is rarely used while doing an application level programming, it makes program confusing, less readable and complex. Also, when this is used, the control of the program won’t be easy to trace, hence it makes testing and debugging difficult, extensive use of “goto” statements make it difficult to impossible to keep the program code in a top down format.

11) Describe some design trade-offs between efficiency and safety in some language you know.

For efficient language design there are many criteria that developers need to keep in mind when creating a new programming language. They usually have to sacrifice one efficiency to achieve safety or vice versa. One of the best examples of these that I was reading about is the one of the dilemma of checking array bounds for any index. For example, JAVA, at the compile time checks that all the reference to arrays have their indices within the bound while on the other hand a language like C does not check for these bounds. This makes C much faster than Java, but also makes it much more insecure, because if you have an error in your code, this is going to be very hard to trace, because you will not get a trace back of the error since it was not check by the compiler on C.

15) How do type declaration statements for simple variables affect the readability of a language, considering that some languages do not require them?

Readability is of prime importance to any programming language. The readability is directly related to cost of development and maintenance in software industry. Since, we know that currently majority of the work in industry is about software reuse and maintenance. This also increases the importance of readability for any language.

But in the case of the use of type declaration statements for simple variables, it may have very little effect on the readability of programs. If a language has no type declarations at all, it may be an aid to readability, because regardless of where a variable is seen in the program text, its type can be determined without looking elsewhere. Unfortunately, most languages that allow implicitly declared variables also include explicit declarations. In a program in such a language, the declaration of a variable must be found before the reader can determine the type of that variable when it is used in the program.