Carlos Villagomez

CPSC 326-01

Assignment 1

**Chapter 1 Reading:**

1. Of all the reasons for studying concepts of programming languages, I find an “increased ability to learn new languages” to be the one that speaks the most to me. I mostly agree with the author when he states, “Once a thorough understanding of the fundamental concepts of languages is acquired, it becomes far easier to see how these concepts are incorporated into the design of the language being learned.” This is very true given how faster it has become for me to pick up new languages now, as opposed to when I was first learning Python and C++ during my Freshman year at Gonzaga. I can still remember the agony of CPSC 122 when I had to make the jump from Python to C++ as there were several adjustments I had to make such as the difference in syntax, pointers, and header files were all changes that I encountered when familiarizing myself when writing programs in C++. However, I can say that learning C++ early on in my CS career has helped immensely in teaching me very similar languages such as C now that I have become more versatile and worked with a variety of different languages over the years such as Python, Java, Assembly, and C++.
2. Readability refers to the ease with which programs can be read and understood, and it serves as an important measure of the quality of programs and programming languages. Next, writability is a measure of how easily a language can be used to create programs for a chosen problem domain. Consequently, many of the language characteristics that affect readability also affect writability. The process of writing a program requires that a programmer frequently rereads the part of the program that has already been written. Finally, reliability refers to when a program is said to perform to its specifications under all conditions.
3. An example of how readability, writability, and reliability can conflict with each other in the context of programming language design is restricted aliasing. This is when you have two or more distinct names in a program that can be used to access the same memory cell. This is considered a dangerous feature in a programming language as it can impede a language’s reliability. Another example of how they can conflict with each other is through exception handling. This is the ability of a program to intercept run-time errors and other unusual conditions detectable by the program, and then take corrective measures. While it is true that in some languages aliasing is used to overcome deficiencies in the language’s data abstraction facilities, there are other languages that must greatly restrict aliasing to increase their reliability. As a result, aliasing along with exception handling both have an impact on a language’s reliability.
4. The relative advantages of compilation, which is when programs can be translated into machine language is that it can be directly executed on the computer, so it has very fast program execution once the translation has been completed. A disadvantage with compilation is that although the machine language generated by a compiler can be executed directly on the hardware, it must nearly always be run along with some other code and most user programs also require programs from the operating system. In addition, before the machine language programs produced by a compiler can be executed, the required programs from the operating system must be found and linked to the user program. Meanwhile, interpretation lies at the opposite end from compilation among implementation methods. In this case, programs are interpreted by another program called an interpreter. The interpreter program acts as a software simulation of a machine where its fetch-execute cycle deals with high-level language program statements rather than machine instructions. As a result, pure interpretation has the advantage of allowing easy implementation of many source-level debugging operations since all run-time error messages may refer to source-level units. However, this method also has its disadvantages such as the execution time is 10 to 100 times slower than compilation systems, a statement, no matter how many times it has been executed must be decoded every time, and pure interpretation often requires more space as well.