CS 408 Evaluation of a Modern Programming Language

Pascal is an imperative and procedural programming language which was designed by a Swiss computer scientist, Niklaus Wirth. The main intentions for this language was to make a language suitable to teach programming on structured basis. The other intention was to develop an implementation of this language which was reliable and efficient. Pascal allows programmers to define their own complex structured data types and build dynamic and recursive data structures. It was useful at the time because programmers were able to create such complex structures like lists, trees, and graphs. Pascal also offered many features like records, enumerations, sub ranges, dynamically allocated variables with pointers and sets. There were many fundamental concepts that Pascal offered which provided it to be a great learning process in terms of systematic discipline. Pascal allows nested procedure on definitions to any level of depth. During the time, Pascal was considered a primary high level language and was used in development for the Apple Lisa and even in the early years of Macintosh. The first impression I got when I saw the blocks of code was that this language was really used for teaching students. The syntax were nearly in English. Wirth’s intention was great because starting off on learning a computer language through English would help the students understand quicker. I definitely think that if it were any other language like C, then the students would have had a hard time understanding.

In all my experience with programming language, I have had a sudden realization that they are structured the same. It is the syntax which make it different or what separates one language from another. Most of the syntax’s and grammar that we learned from our previous programming courses were similar between these languages. There were subtle differences that made it different. For example, comments in Pascal are placed between braces. However, if we wanted to comment in C++ we would place a slash and a star afterwards. There were other differences that I found to be convenient like the case insensitive features. This is just one of my pet peeves because having to be able to write variable names through different cases is just confusing. Not only confusing in readability but also in writability. This can cause problems later, especially if it is a large program. One problem can lead to various other problems. As I was working around with Pascal, I learned that the mastery of this language was trying to teach the programmers good habits and practices. If there were things that the user wanted to get done, there was a way in the language to do it but only in one of the ways. There were no shortcuts to get around certain things. In Pascal, it was teaching the users to do everything the right way or the compiler would not accept the code that was displayed. There were benchmarks that showed the performance of C++ and Pascal, and the graphs showed that Pascal was just as powerful as C++. Although, C++ was a little bit faster in performance, Pascal was just as powerful and programs are actually more readable and maintainable than it is for C++ or C. Writing programs is one thing, but compiling is another. The comparison between Java and Pascal was different by a lot. Pascal compiled quickly compared to Java. In Pascal, we don’t need to worry about features like namespace because each unit has its own identifier. Unlike C++, where every identifier needs to be unique and throughout the entire program. When playing around with the code, I needed an IDE to work with and there was one called Free Pascal. Using an IDE, I was able to save huge amounts of time and be able to compile my program. The other cool thing was that, Pascal offers object oriented programming in later editions like the Object Pascal.

Teamwork plays a huge role in any group project. The main concerns are the agreement between both team members. I was able to learn a lot through this group project and what it means to be able to cooperate with one another. With team work comes great responsibility. There were many things that we were able to accomplish like working on the code, learning the language, learning the different features, etc. This all would not have happened if it weren’t for the continuous communication between the two partners. I think that most of the time spent was trying to communicate on a normal basis on what and how we should approach certain things. Looking around different programming languages, I was able to find Pascal because of its well organized and structured programming style. I suggested that we work on Pascal and there were continuous communication between us on what were the advancements of the features. One of the major steps, was to actually learn the language. Starting from the syntax to the semantics, we had to learn it all. It took us time about two weeks or so. I played around with the code a bit to see what were some of the major features and interesting things that this programming language had to offer.

Overall, Pascal is a high level procedural programming language which is recommended for beginners as it contains fundamentals for newcomers. The syntax and semantics of the code is easy for anyone to understand right away. It is a very easy programming language and helps understand the basics of the world of programming. It’s also a stepping stone to learning the C programming language. The transition from Pascal to C will become a lot easier. There are even times that Pascal can be used for technical problems. As we have seen from the benchmarks, Pascal outperforms some languages in different aspects. Pascal may be an outdated language and there are other languages that are more widely used but Pascal still has some uses.