1. What was the focus and application of your project?

The focus of my project was input/output processing. The application was a new language interpreter.

1. How did you meet, exceed, or fall short of your original expectations?

I accomplished everything I proposed, including all extras. I have a working interpreter that handles valid input with no ambiguity and can do exciting extras like recursion and useful scoping

1. What changes were made to your project during the process?

I had to reorder items on my schedule because I didn’t anticipate they would be as intertwined as they were. This required me to put in many extra hours to make up the difference as I underestimated the impact of implementing these things.

1. What were the most important/interesting lessons learned about the focus?

It all starts at the bottom. Creating output for valid input is a non-trivial task. There can be so much ambiguity introduced since we’re dealing with humans. Clear rules need to be defined to weed out this ambiguity.

1. What were the most important/interesting lessons learned about the application?

It takes a lot of time and knowledge to make an interpreter. The lexer and parser are critical to the rest of the application. Even though these components consist of seemingly simple-minded tasks, they shape everything higher up. Design is always critical in software development. This is especially true in this type of application where the whole thing is extremely cohesive. Waterfall would be a very good fit for this type of thing.

1. What were the most important/interesting lessons learned about yourself?

I can do difficult things. I can go out, research, and implement whatever I set my mind too, providing I put in the work. Sometimes we have to put in extra hours, but if we want something badly enough we can get it done.

1. Final Thoughts

This was a great project. I’m glad I got to do something I’m interested in and I found rewarding. If I had to do it again, or make a big release, I would do a few things differently. I would try using recursion more. And clock it against other possibilities for components. I would start with a problem; something that bothers me about development, and then I would create a language that makes it really nice. It would be a language with a purpose, not just another language. I would design the whole thing first so I could minimize surprises during development.