**CS323 Assignment Documentation**

***<should consists of about 2-3 pages>***

1. **Problem Statement**

How do we write a syntax analyzer for the programming language Rat17F?

1. **How to use your program** 
   1. Unzip Assignment3
   2. Follow path in folder provided ( \Assignment3\Assignment3\Debug\ )
   3. Add Rat17F source code text file in current directory.(Be sure to name the text file Rat17F)
      1. By default there will already be one called “Rat17F”
      2. If adding a new source file make sure to name it Rat17F.txt
      3. It would be easier to open the “Rat17F” text file and paste your test case code in there and save it.
   4. Once file added or using default file, right click on “Assignment3” the “Application” under (Type column) and click “run as administrator”.
   5. Application will now run.
      1. There might be a window that pops up saying “Windows protected your PC”, click on “More info” and then click on “Run anyway” near the bottom of the window.
   6. Once done go back to the path (Assignment3\Assignment3\Debug\)
      1. There will be 2 text documents that are named “instructionTable” and “symbolTable”. “instructionTable” will have the output of all the assembly instructions. “symbolTable” will have the symbol table of course and also if an error happens it will be outputted here as well. Also keep in mind if an error happens the program will terminate and write to this file “symbolTable”.
2. **Design of your program**

Used regex library to extract would be tokens from Rat17F source code file and pushed them into a vector, then used a switch statement that’s argument was a string and depending on the case that executed that certain current state was returned and this process continued until the end of the string. Once reach at the end of the string a function will be called to determine whether the string of characters or character is accepted or rejected. Once the lexical analyzer is done the syntax analyzer starts, the syntax analyzer was used implementing a Recursive Descent Parser. We Created functions for every non-terminal symbol, the parser starts by calling the Rat17F() in the main of the program. Now that the Rat17F function has been called it will go through each token and lexeme we have entered two vectors to parse all the way through the source code. IMPORTANT: The output of the program (parsing functions called and errors) will be outputted to a text file called output.txt. The output will show each function/production that is used to analyze the tokens and lexemes.

1. **Any Limitation**

*None that we are aware of.*

1. **Any shortcomings**

*The main shortcoming for this assignment was not being able to properly implement the type matching functionality.*