**COVER PAGE**

# **CS323 Programming Assignments**

**Fill out all entries 1 - 7. If not, there will be deductions!**

1. Ahzaz Laeeque [Thursday 7pm]

2. Morgan Heaslet [Thursday 7pm]

3. Anthony Comito [Monday Wednesday 11:30AM]

2. Assignment Number [ Assignment 1]

3. Due Dates **Softcopy** [ 10/07/2018], **Hardcopy** [ 10/11/2018]

4. Turn-In Dates **Softcopy** [10/07/2018], **Hardcopy** [10/11/2018 ]

5. Executable Filename [ ]

(**A file that can be executed without compilation by the instructor**)

6. Lab Room [ CS101 ]

**(Execute your program in a lab in the CS building before submission)**

7. Operating System [Windows]

**To be filled out by the Instructor:**

GRADE:

COMMENTS:

**CS323 Documentation**

About 2-3 pages

1. **Problem Statement**

Basically, this program is going to create a lexical analyzer for Rat18F. It will be going to take each token and then identifies each of them if there is an integer, real number, separator, operator, or a keyword.

Input: File containing Rat18F

Output: File identifying the token type

1. **How to use your program**

In the CS101 lab, use the command prompt and run the file .

1. **Design of your program**

We used some C++ libraries to classify the lexeme with it details like name and number. Every token has its own value in the dictionary. For example, token name, Lexeme name.

I will explain a little bit. My lexer is going to take a string and then each token in that string is going to check for integer, real number, space, identifier, separator and operator. After checking take each token and check it type and then send the type to FSM and add the token to the Lexeme string. Once it broken by a separator or whitespace then it checks the FSM to see if it is in a valid state. If it is then prints the lexeme as a valid token if not than delete that lexeme and start continuing again from where it left.

1. **Any Limitation**

the current limitations are the fact that keywords are hard coded and not received from a file list of keywords, so there may be many more we don't know are keywords yet that the program won't recognize

Also, the lexer outputs each lexeme to a file, but we don't current store the token object for later use or return the token back to the main program.

1. **Any shortcomings**

NO