COVER PAGE

CS323

Programming Assignments

Peer Review (Check one)

1. Names [1.Jacqueline Isabel Cardenas], (ThumbUP [X] or ThumbDown [])

   [2. Daniel Pestolesi], (ThumbUP [X] or ThumbDown [])

   [3. Garrett Reeve], (ThumbUP [X] or ThumbDown [])

2. Assignment Number [Project 1]

3. Turn-In Dates: **Final Iteration with Documentation** [February 25th]

4. Executable FileName [lexical]

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

5. LabRoom [CS 104, Tuffix distribution]

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

6. Operating System/Language [C++]

To be filled out by the Instructor:

GRADE:

COMMENTS:

CS323 Documentation - Project 1

Group Members: Jacqueline Isabel Cardenas,

Daniel Pestolesi, Garrett Reeve

Problem statement

For the first assignment a lexical analyzer was designed that identifies each character. The program reads a file, identifies each character and write out the results to an output file. The characters that the program can identify are keywords, separators, operators, identifiers and numbers.

A couple test cases would be:

! Declare and assign a number !

int number;

number = 9;

output:

Tokens:		Lexemes
Keywords	=	int
identifier	=	number
separator	=	;
Identifier	=	number
Operator	=	=
Integer	=	9
Separator	=	;

! Do some math !

```
int firstval;

int secondval;

firstval = 3;

secondval = 4;

firstval * secondval;
```

Output:

Tokens:	=	Lexemes
Keywords	=	int
Identifier	=	firstval
Separator	=	;
keywords	=	int
Identifier	=	secondval
Separator	=	;
Identifier	=	firstval
Operator	=	=
Integer	=	3
Separator	=	;
identifier	=	secondval
Operator	=	=
Integer	=	4
Separator	=	;
Identifier	=	firstval
Operator	=	*
Identifier	=	secondval

Separator	=	;

! count to 5 !

1;

2;

3;

4;

5;

Output:		Lexemes
Integer	=	1
Separator	=	;
Integer	=	2
Separator	=	;
Integer	=	3
Separator	=	;
Integer	=	4
Separator	=	;
Integer	=	5
Separator	=	;

How to use your program

Tuffix was used for this assignment. Open terminal from the file lexical.cpp compile program by typing.

“ g++ -o lexical lexical.cpp” ; Executable also provided so this step may be skipped.

Execute by typing ./lexical

Input the text file desired to be analyzed. Sample input text file provided.

Output the text file you want the analysis to be output into. Sample output text file provided, all d

It will then output the information as demonstrated in the problem statement to the desired output text file as well as in the terminal.

Design of your program

Char arrays were used to store different identifiers which are keywords, separators, operators. Bool was used in detecting different states. And the identification of comments was done with Boolean and an array to be able to store block comments !.

Useful Algorithms used

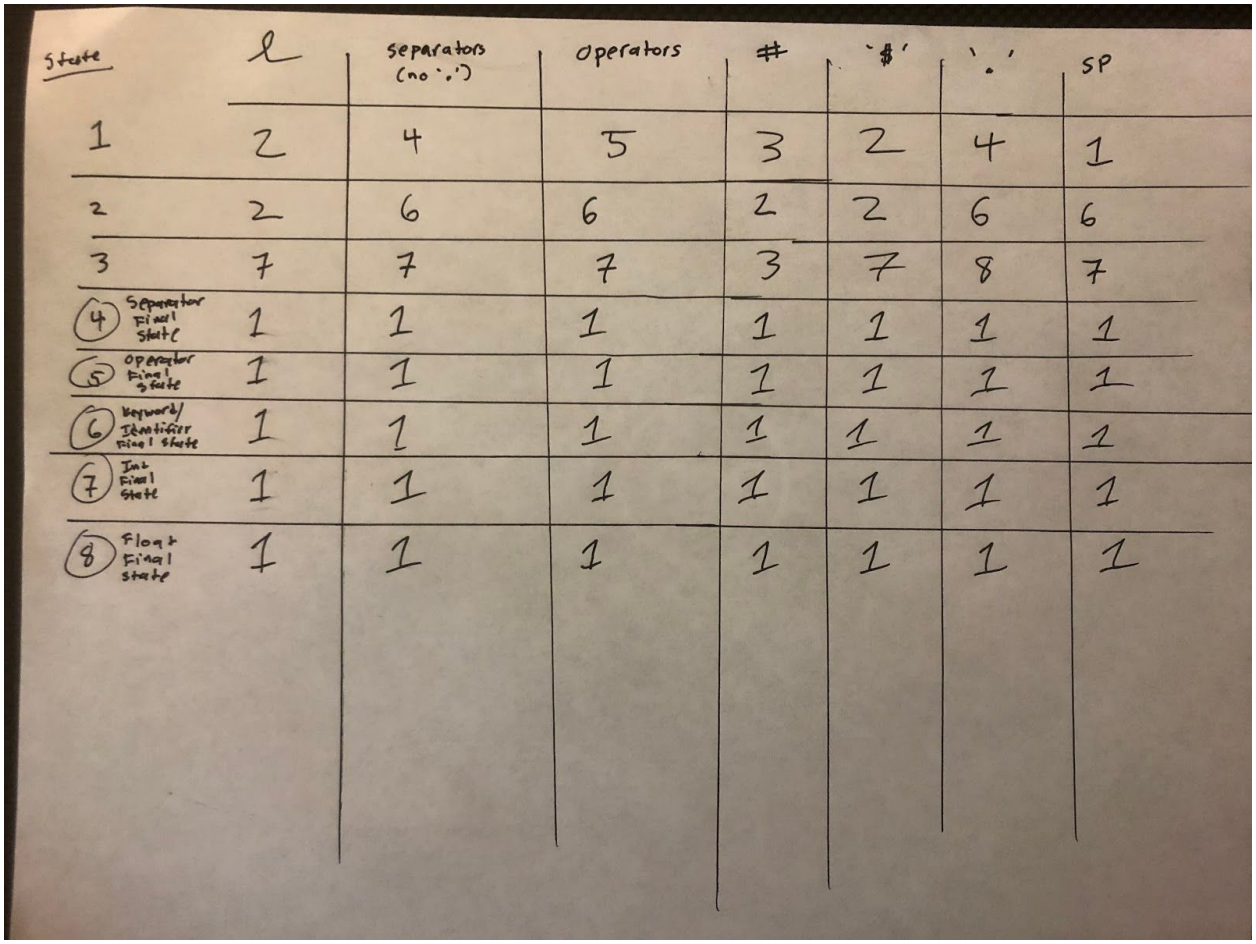
Isdigit checks whether value is a digit character

Isalnum checks whether value is alphanumerical meaning uppercase, lowercase or number

Any limitations

Program is only capable of identifying predefined characters.

Program does not identify floating point numbers.



State	l	separators (no '.')	operators	#	'$'	'.'	SP
1	2	4	5	3	2	4	1
2	2	6	6	2	2	6	6
3	7	7	7	3	7	8	7
(4) Separator Final State	1	1	1	1	1	1	1
(5) Operator Final State	1	1	1	1	1	1	1
(6) Keyword/ Identifier Final State	1	1	1	1	1	1	1
(7) Int Final State	1	1	1	1	1	1	1
(8) Float Final State	1	1	1	1	1	1	1

Finite State Automata Table