**Kennesaw State University**

Concepts of Programming languages

Professor: Betty Kretlow

Student: Sams Khan

**Module\_3 – 1st Deliverable Report**

**TABLE OF CONTENTS**

Work Performed………………………………………………………………...Page 2

Input file Screen shot...…………………………………………………………Page 3

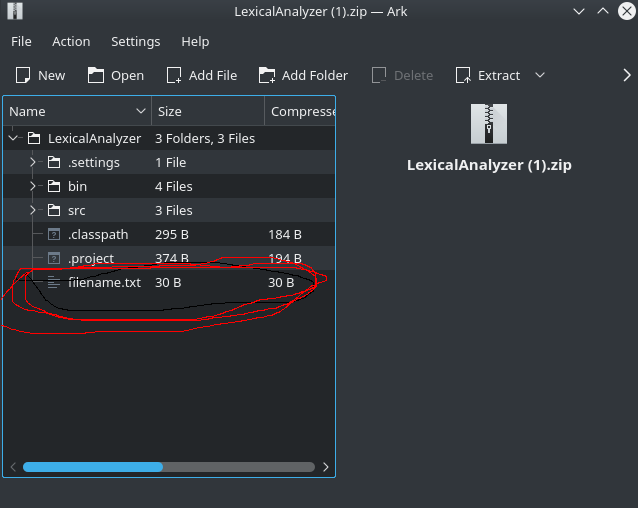
Output file Screen shot…………………………………………………………Page 4

**Work Performed**

* First I started off watching Dr Gaylers video on Module 2.
* I got a general idea on how the scanner should work, basically it must be able to identify the symbols based on the given grammar.
* First I knew I had to store the types of tokens that I am to define which is why I created the TokenType.java file, there I made an enum and listed out all the different identifiers for the symbols.
* Then I needed to be able to define what I was going to display, so I made a class specifically for tokens called Token.java
* My token class takes in the row number column number, lexme and the tokentype, this class also consists of some getter methods that return each of the parameters its been given.
* Now that I had the definition of what a token is and how it should be defined, now I have to describe how it should be analyzed.
* The next class created was the lexAnalyz.java. Here in the constructed the only input parameter is the file which is defined as “filename.txt” it can be found in my “Lexical Analyzer” it should be in the same folder as my “src” folder.
* In the lexAnalyz constructor I scan in the sample lua code.
* Then I have a method called processLine that takes each line of code from the file and adds the token to a list of tokens and also includes th line number and increments the index of the list.
* Then my getTokenType defines what token actually correlates to which symbol.
* The allditgit method checks if the string entered is actually a number or not, this helps to identify numbers for the parser
* getLexeme method returns me the actual symbol that correlates to the token
* skipwhitespace just lets me not scan white spaces

**Input filename**

* **Location**

The input files name is “filename.txt”

**Output file**

