**Assignment 1 Documentation**

G - 19 ChonkskyNormalForm

**Team Members:**

Manisha Katariya 2017B3A70354H

Aditya Agarwal 2017B1A71075H

Jajal Bansal 2017B3A71610H

Rudrajit Kargupta 2017B3A70452H

**Test files and their outputs:**

3 test files called "inp1.txt", "inp2.txt", "inp3.txt", and their output files have been included in the zip file. The corresponding output files are "out1.txt", "out2.txt", "out3.txt".

**Language Description:**

Our language includes the following keywords:

* while (while loop)
* if (if statement)
* else (else statement)
* void (void type identifier)
* @ (String type Declarator)
* # (Character type Declarator)
* $ (Integer type Declarator)

Our language includes the following operators:

* + (addition)
* - (subtraction)
* \* (multiplication)
* / (division)
* % (modulo operator)
* ++ (increment operator)
* — (decrement operator)
* ~ (negation operator)
* > (greater than)
* < (less than)
* == (equality)
* = (assignment)
* . (string concatenation)

The alphabet accepted by our language includes the symbols described above along with all lowercase and uppercase letters, digits from 0 to 9, (, ), {, }, ; as the end of statement indicator and ^ as the comment indicator. All other symbols are illegal.

**The output format for our Lexer:**

<<Token type>, <identified lexeme>>  on line number <line number>

For example: The statement $ num = 0; on the fourth line of the program produces the following output:

          <Integer type declarator, $>  on line number 4

|  |  |
| --- | --- |
|  | <Identifier, num> on line number 4 |
|  | <Assignment operator, =>  on line number 4 |
|  | <int\_literal, 0> on line number 4 |
|  | <End\_of\_statement representor, ;>  on line number 4 |