CS 335 Semester 2019–2020-II: Assignment 1

13th January 2020

Due Your assignment is due by Jan 26 2020 11:59 PM IST.

General Policies

- You should do this assignment ALONE.
- Do not plagiarize or turn in solutions from other sources. You will be PENALIZED if caught.
- We MAY check your submission(s) with plagiarism checkers.

Submission

- Submission will be through Canvas.
- Create a zip file named "`cs335_<roll>.zip`". The zipped file should contain a folder `assign1` with the following files:
  - Implementation files in your chosen implementation language.
  - Four test case files containing proper non-trivial Java programs. You should name the test files as "`test_<serial number>.java`".
  - A script file named `run.sh` similar to the sample "`run.sh`" shared with this assignment problem. The script should generate an output file "`test_<serial number>.out`" with the desired output format, corresponding to an input file.
  - A `PDF` file describing any tools that you used, and include compilation and execution instructions.
- You SHOULD USE LaTeX typesetting system for generating the PDF file.
- Submitting your assignments late will mean losing points automatically. You will lose 20% for each day that you miss, for up to two days.

Evaluation

- Please write your code such that the EXACT output format is respected (if any).
- We will evaluate your implementations on a Unix-like system, for example, a recent Debian-based distribution.
- We will evaluate the implementations with our OWN inputs and test cases, so remember to test thoroughly.

Problem 1 **[50 points]**

Create a lexer (i.e., scanner) for the Java language. The complete lexical structure for Java 8 is available here[1].

The output of the lexer should be a file containing a list of the form **Lexeme | Token | Count**.

[1] `https://docs.oracle.com/javase/specs/jls/se8/html/jls-3.html`

Example

Input. Consider the following input Java program.

```
public class Program {
    /*
     * This is my first java program.
     */

    public static void main(String[] args) {
        int data = 50; // declaration
        boolean flag = false;
    }
}
```

Expected output. The output should (1) list and classify all unique lexemes into proper syntactic categories, and (2) provide a count of the ⟨lexeme, token⟩tuples.

Lexeme	Token	Count
public	Keyword	2
class	Keyword	1
Program	Identifier	1
{	Separator	2
static	Keyword	1
void	Keyword	1
main	Identifier	1
(	Separator	1
String	Identifier	1
[	Separator	1
]	Separator	1
args	Identifier	1
)	Separator	1
int	Keyword	1
data	Identifier	1
=	Operator	2
50	Literal	1
;	Separator	2
}	Separator	2
boolean	Keyword	1
flag	Identifier	1
false	Literal	1

The tool should report any tokenization error due to lexical errors in the input program.