LAB ASSIGNMENT 9: MAKE A LEXER

MAXIMUM POINTS: 12 DEADLINE: JANUARY 13TH, BEFORE LAB

1) Write a JFLEX specification containing regular expressions for the tokens necessary for your language specification. Use the scanner generator to produce a scanner.

The scanner shall produce as the output a table in the following format:

```
Summary:
     Tokens          <number of tokens>
     Identifiers     <number of identifiers>
     Literals        <number of literals>
     Keywords        <number of keywords>
     Operators       <number of operators>
     Separators      <number of separators>
```

Each appearance of the items is counted, also duplicates!

2) The scanner also shall print a simple symbol table of identifiers as well as literals in the following format.

```
Identifiers:
     identifier1
     identifier2
     …
     identifierN
Literals: (with type information in parenthesis)
     "constant1" (String)
     87921 (Integer)
     …
     7.281 (Float)
```

These lists must not contain duplicates and shall appear sorted!

Make use of the Java standard libraries for the necessary data structures.

3) The scanner should report lexical errors, specifying the location in the source code file.

OUTCOME

Submit an archive called **`Lab10.Surname.Name.ZIP`** which contains:

a) The lexical specification file **`minilang.jflex`**
   Make sure JFlex/Java generates the scanner class called **`SymTabs`**
b) The source code of your mini-program as **`Main.java`**
c) The source code of 2 mini-programs with lexical errors (with a comment in the line above to the error) as **`MainL1/MainL2.java`**
d) A **`compile-run.sh`** to run the lexer-generator and compile the code and run the scanner.

Note: Non delayed submissions will be allowed!

Note: Non-compliance to these requirements will cause -25% of points.

Remember: Submission of replica of one colleagues' work harms both students!

Suggestion: Be creative, write something unique ;-)