

## COP3402: Spring 2011 (2 part assignment)

### Assignment # 1

#### 1. Using my Pascal-S lexical analyzer written in Pascal

<http://www.cs.ucf.edu/courses/cop3402/spring2011/Assignments/PascalSLex.pas>

and the Pascal-S grammar

<http://www.cs.ucf.edu/courses/cop3402/spring2011/Assignments/PascalSGrammar.doc>

as requirements documents, write a lexical analyzer in C or C++ that does the same. You are free to format output differently, but at least the information displayed by PascalSLex.exe must also be displayed by your program. Do NOT use some existing tool to convert Pascal to C/C++. Your code should be clean and that will not come out of a Pascal to C/C++ translator. Turn in a C or C++ program (source) that runs on a standard C/C++ compiler (no Windows sugar; just a console application). Name your source with your First Initial followed by your Last Name followed by "Asn1\_1" and the C or C++ extension. E.g., Remo would be RPillatAsn1\_1.c or perhaps RPillatAsn1\_1.cpp

#### 2. Using part 1 as a starting point, extend your lexical analyzer so it accepts:

- a. Comments that start with { and end with }. Thus, a comment can appear as { my comment } or (\* my comment \*)  
Nesting is not meaningful, so { { (\*\*) } is a single silly comment (the opening brace { and closing brace } are all that matter).
- b. Allow directives in comments. A directive is a single letter (cap or lower case) followed by a plus (+) or minus (-). Blanks can separate directives (or not) but they cannot separate a directive letter from its + or -. Once anything is encountered other than a directive, the rest of what is inside the (\* \*) or { } is a comment. So, (\* A+ B-C+Happy \*) contains three directives followed by the comment Happy. Directives should be reported just as lexical items are. Be careful of things like (\* A+B\*)
- c. Accept the percent sign (%) as equivalent to the reserved word MOD, and double divide (//) as equivalent to the reserved word DIV.
- d. Accept ++ and -- as two new operators that are associated with the new tokens incsy and decsy. Although semantics are not meaningful yet, these are intended to be the C increment/decrement operators.

Turn in a C or C++ program (source) that runs on a standard C or C++ compiler (no Windows sugar; just a console application). Name your source with your First Initial followed by your Last Name followed by "Asn1\_2" and the C or C++ extension. E.g., Remo would be RPillatAsn1\_1.c or perhaps RPillatAsn1\_1.cpp

**Due Dates:**

**Part 1 is due on Tuesday, February 1 before the end of day (11:59PM)**

**Part 2 is due on Tuesday., February 8 before the end of day (11:59PM)**

**Turn in must be done via Webcourses. Remember, only the source is to be turned in. We will provide some test cases by next week, but we will also run many other test cases as part of the grading.**

**Aids:**

**If you wish to actually run and play inside the Pascal system, you can download a compiler and ide at <http://www.bloodshed.net/devpascal.html>**

**I also included two sample PascalS programs in the Assignments directory. You could even run the PascalSlex.pas file as a test case.**