

CS350 Lab 1: Lexical Analysis

Use Eclipse to obtain the `tiger` project, and check the definition of `which_cs350_lab` in `tiger.cc` to make sure it has the value 1 (you'll be changing this in later labs; if it is anything other than 1 now it will cause confusion).

Modify the file `tiger-lex.1` (and, if necessary, other files) to make a complete lexical scanner for the “tiger” language described in the appendix of Appel’s book, *except* that you do not have to allow nested comments. Your lexical analyzer should process special characters in strings, and detect errors involving unclosed comments or string literals (by creating regular expressions to match these kinds of illegal input, with associated actions that print errors). A few more details about lexical analysis for tiger are described in the programming exercise section of Chapter 2 of Appel’s book, but the descriptions of files do not correspond to our C++ files. A correct lexical scanner should pass all the tests triggered when you run the `Run-Tests` script included with the tiger project (see below).

In addition to properly commenting your program and using clear names, you should record important design decisions for this lab in the file `Lab1-LexicalScanning` in the `Design_Documents` subdirectory. Think of this as a place to record things you wish you had known before you started the lab — it can include both things that you did in a particularly interesting way, things that took some time to figure out, or even things that you didn’t do but would have done if you had thought about them earlier or had time to work more.

For 75% partial credit, do not bother with processing special characters in strings or printing clear messages for errors — your lexical scanner will still be sufficient to go on with the later exercises if you do this.

In general, you should be doing incremental writing and testing of your projects. For most labs (including this one) you may want to do this by trying to complete the partial-credit level of each lab and make sure that works, and then go on to the full-credit version. Remember that we recommend you submit your work each time you reach a significant “milestone” of development (as well as at the end, of course).

For 10% *bonus* credit, figure out how to make flex recognize “nested comments”. This may cause a change to some of the test results, so get everything else working first and check it before you attempt this.