

# COMP 520 Compiler Design

## Individual Assignment #1

Scanning and Parsing MiniLang

Due: Wednesday, January 25th, 11:59pm

### Overview:

The purpose of this assignment is to get everyone familiar with scanning and parsing, and to experience some of the tools on a simple project.

You may use any tools you wish (for example, flex/bison in C, or SableCC in Java). Many tools are listed at [http://en.wikipedia.org/wiki/Comparison\\_of\\_parser\\_generators](http://en.wikipedia.org/wiki/Comparison_of_parser_generators). You may also do a hand-coded scanner and/or parser, although this will mean that you will not get experience using the tools that you might use in your group project. To simplify the grading and work for the TA's, please ensure that your compiler will run on the Trottier machines.

The MiniLang programming language was initially defined by Vincent Foley, and the details of the syntax will be decided in class. You can find an example program, and the specifications on the course website and Facebook group.

### Question 0: *Name the COMP 520 dragon*

(the winning name gets +1 points) What is your suggested name for the dragon on the COMP 520 home page, <http://www.cs.mcgill.ca/~cs520/2017/> ?

### Question 1: *Correct Example Programs (5 points)*

Develop two correct programs, stored in files with a suffix of `.min`. Try to make each program perform a real computation, and try to use as many MiniLang language features as possible.

### Question 2: *Syntactically-incorrect Example Programs (5 points)*

Develop five small programs, where each program exhibits a different scanning or parsing error. We will collect these together, along with all of the other students' tests, into one big torture test suite which we will use for testing your scanner/parser.

### Question 3: *Scanner/Parser for MiniLang* (20 points)

Implement a scanner/parser for MiniLang. For the purposes of this assignment the scanner/parser will be used to decide if an input program is “VALID” or “INVALID”. Thus, if the input program is syntactically valid it should print on stdout the string “VALID”. If it is invalid it should print on stdout a string starting with “INVALID”, followed by the error message. A standard run script will be distributed by the TAs to ensure your input and output formats conform to a common specification.

For the purposes of debugging, you may want to implement a `-debug` flag which enables tracing of the scanner and parser. This is optional.

### What to hand in

On myCourses, hand in a tar.gz file of the form `firstname_lastname.tar.gz`.

The structure of the files inside that tar-ball should be:

```
firstname_lastname/
  README      (Your name, student ID, any special directions for the TAs)
  DragonName.txt (Your answer to Question 0)
  programs/
    valid/    (your two correct programs)
    invalid/  (your five syntactically incorrect programs)
  src/        (the source code and build files)
  run.sh      (an updated run script based on the example given from the TA)
```

It is very important to have this exact structure, so that we can run our tests with scripts. If you do not conform to the run script specifications then you will have to resubmit your assignment.