

CS320 Assignment #3

Purpose

This assignment is designed to familiarize you with LUA programming, parsers.

Requirements

This assignment consists of one major requirements.

- 1) Developing solutions to the problems below.
- 2) Your assignments will be checked, submitted and graded electronically.

Problem

- 1) prog3_1.cpp will be implemented in C/C++. Program 1 will take a single command line argument (other than the name of the program). The command line argument will be the name of a lua file. Your program should then execute the Lua file in a lua environment you create in your C/C++ program. You should expect the lua-5.3.4 source folder is in the current directory. (Please do not commit the lua-5.3.4 folder to your repository).

Program 1 will be compiled like so:

```
g++ prog3_1.cpp -o prog3_1 -I lua-5.3.4/src -L lua-5.3.4/src -l lua -l m -l dl
```

prog3_2.lua In Lua, implement the Infix to Postfix function that went over in class. The function InfixToPostfix(str) takes a single argument (it will be an input string). Tokenize that input string by space (split by space) and then apply the infix to postfix algorithm to return a postfix string.

prog3_3.cpp will be written in C/C++. It will be called prog3_3.cpp and can/should be an extension of prog3_1.cpp The program will create a lua environment, load/run the file specified by the command line argument. It will then take in a line of input from the command line, call the InfixToPostfix() function in lua (use the dostring), retrieve the resulting postfix string from the lua stack (use the checkstring function) and then print the resultant postfix string.

- 1) After you have committed your files you will have to grant me developer access to your repository. To do this you must open the project on gitlab.com, and open the Members page that is under the settings menu on the right hand side of the page (It looks like a gear). Add me to your project by typing in my username: slindeneau and make sure that the project access is set to developer.

You will need to do this for every assignment.

- 2) The last step involves verifying that everything is working correctly. Please go to:

<http://cs320.lindeneau.com>

Input your first and last name (as it appears on blackboard), your gitlab username and select the assignment you would like to grade. Verify that all of the parts of your program work as expected.

DO NOT ENTER ANY PASSWORDS

Passwords are not required for cs320.lindeneau.com to function. If your project does not get graded, you have not given correct access to the correct developer account on gitlab.

Additional Details

- You should be able to investigate any issues you have on your own and spend the time necessary to understand what is happening overall.

Late Policy

Late programs will be accepted with a penalty of 5% per day for seven days after due date. Turn in time will be determined by either date of rubric submission OR timestamp on graded files, whichever is later.

Cheating Policy

There is a zero tolerance policy on cheating in this course. You are expected to complete all programming assignments on your own. Collaboration with other students in the course is not permitted. You may discuss ideas or solutions in general terms with other students, but you must not exchange code. (Remember that you can get help from me. This is not cheating, but is in fact encouraged.) I will examine your code carefully. Anyone caught cheating on a programming assignment or on an exam will result in a zero for the class or assignment.