CS320 Assignment #3

Purpose

This assignment is designed to familiarize you with Lua while working with C/C++.

Requirements

This assignment consists of five major requirements.

1) Developing solutions to the problems below.
2) Ensuring that your programs compile and run correctly using the tools available in a standard linux distribution.
3) Documenting your solutions correctly.
4) Organizing your *git repository* correctly.
5) Running your solution on the autograder.

Problem

1) You must develop solutions to the problems below. We will be using the solutions from earlier assignments in later assignments.
2)
    a. Program 1 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
    ```

    b. Part 2. 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. This should be in prog3_2.lua
    c. Program 2 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 stdin, 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.

3) You must ensure that your code can be run on a standard Debian based linux distribution using the shell tools of your choice. You may develop your solution on any machine you desire, as long as the final solution works on a standard linux distrubtion.
4) Your solution must have a complete comment header as is detailed below. During runtime, each of your solutions to section 1 must output a correct **title string** as the first line printed. It should be in this format:

   `Assignment #1-1, <NAME>, <EMAIL>`

   Each problem should follow this format, with the second number incremented for each problem in part 1. For example the second problem in part one should have the title string:

   `Assignment #1-2, Scott Lindeneau, slindeneau@gmail.com`

5) You must place a copy of your solutions inside a repository named cs320Assignment1 in your git profile on gitlab. Your files **MUST** be named **`prog3_1.cpp`**, **`prog3_2.lua,prog3_3.cpp`** and must not be modified after the turn in time. The modified timestamp for each file in your **`repository`** will be used as the submission time for that file. If it is after the due date, it will be counted as late.

   You are also responsible for two additional files that will be part of every assignment (after assignment #1). You **MUST** place a copy of the programming rubric in your repository. You must not rename it (it should be: cs320programmingrubric.pdf) and by placing it in the repository you should also be reading it and acknowledging that this is the rubric by which you will be graded.

   You will also place a README.md file[1] in your assignment directory. This file should contain a description **IN YOUR OWN WORDS** of the project along with a short description of each file, what it is, how to compile and/or run it. These descriptions may be short as long as they are accurate.

[1] If you are familiar with reddit commenting/posting then you are familiar with markdown. They use markdown exclusively for user provided content. Most git repos use md as the format of choice for readme files and install text.
Here is a good cheat sheet: https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet

CS320 Fa2017 Assignment #3
DUE: 11/22 11:59pm

YOU MAY NOT HAVE ANY OTHER FILES IN YOUR GIT REPOSITORY. If you have any other files in your git repository the assignment will be considered **not** complete.

Additional Details

- Should the max length input be left out of a program description, assume a reasonable value considering previous problems.
- Multiple spaces may occur. There may even be spaces before a newline character without a token.

Late Policy

Programs turned in by the due date will receive 120%, programs turned in 7 days late will be worth 100%, after 7 days 0%.

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 receive an "F" in the course, and a referral to Judicial Procedures.

CS320 Fa2017 Assignment #3
DUE: 11/22 11:59pm