EECS 268: Lab 03: Stacks

In this lab, you have to make a program which can run in two modes:

1. Test mode
2. Parser mode

1. Test Mode:

You enter Test mode by writing “-t” in command line arguments while running the program
For example: . /prog -t

Test mode will be responsible for testing the functionality of stack i.e. whether it is set up correctly or not. For this, you run a battery of tests to check mainly, the push, pop and peek functionalities of your stack. You may have more tests to cover corner cases.

Your stack should PASS these four REQUIRED TESTS: (1) New stack is empty; (2) Push on empty stack makes it non-empty; (3) Popping all elements from the Stack makes it empty; (4) Popping an empty stack returns warning.

Here are some example of extra testing conditions:

1. Peek matches last pushed value
2. Most recently Pushed value is on top
3. Push 4 values and check whether they are in proper order.

For implementing test mode you will require a StackTester class which runs the series of tests independently i.e., new Stack is created for each test. Each test should print “PASS” or “FAIL” according to the conditions. You can run all the tests by calling them from a single test function.

StackTester.h has been given in wiki page. You can add more method into this class.

Here is one method example of StackTester.cpp class, which you can take as a reference.

You have to at lease add two extra test cases for getting full points.

```
void StackTester::test01()
{
   charStack test;
   std::cout<<"Test #1: Newly created stack is empty: ";
   if(test.isEmpty()==true){
    std::cout << “Pass”;
     }
   else
     {
   std::cout << “fail”;
  }}
```

2. Parser Mode:

You can run parser mode by writing “-p” in command line arguments when running your program
For example: . /prog -p

Parser mode works as an application of stack. You have to ask user to enter a sequence of characters. Your program (in parser mode) checks whether the sequence of parenthesis is balanced or not. Some examples of balanced and unbalanced sequences have been given on the wiki page.

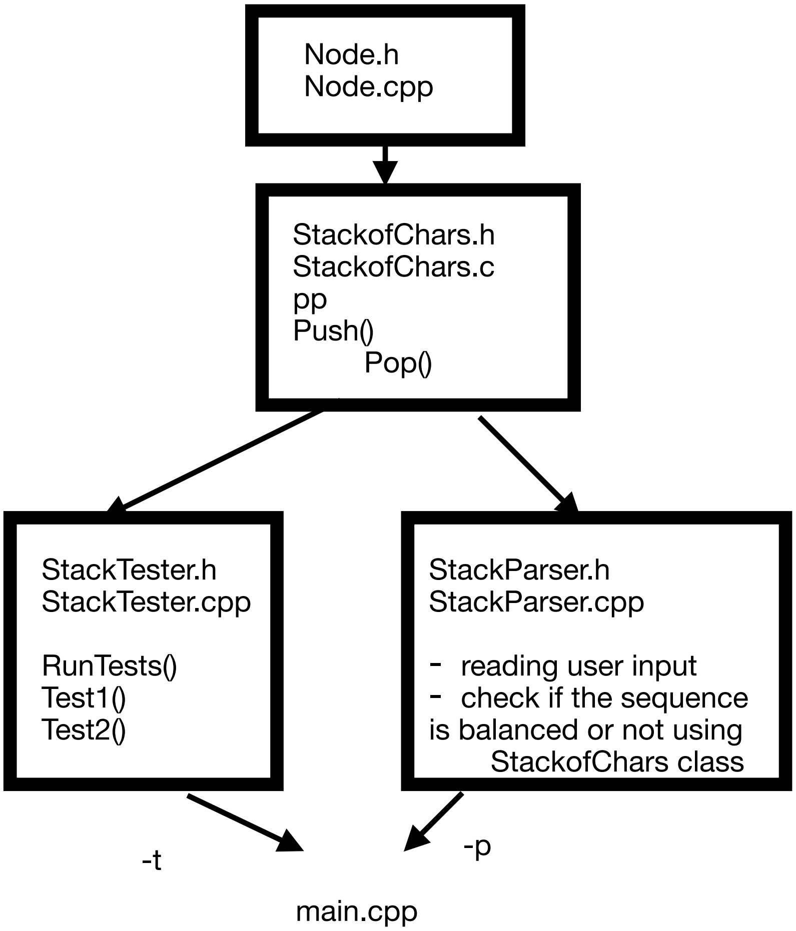
To run your program in parser mode, you will be required to make a Parser class which is responsible for reading the user input (string) character by character and checking whether the string is balance or not by using the functions of StackofChars class.

Hint: You will be using push and pop function according to the “{” and “}” respectively.

List of Files in your tarball:

Node.h
Node.cpp
StackOfChars.h
StackOfChars.cpp
StackTester.h
StackTester.cpp
StackParser.h
StackParser.cpp
main.cpp
Makefile

Here is the structure flow:


Node.h
Node.cpp
StackofChars.h
StackofChars.c
pp
Push()
Pop()
StackTester.h
StackTester.cpp
RunTests()
Test1()
Test2()
StackParser.h
StackParser.cpp
- reading user input
- check if the sequence
is balanced or not using
StackofChars class
-t
-p
main.cpp