Fangshuo Li

Kostas Alexis

CS302

9/16/20

PA01

StackLinked.h

Explicitly did not create a separate cpp file for simplicity and I didn’t see it specified in instructions. To make one I would just create a new file and link it to each of the builds in the makefile.

1. Param ctor
   1. Not much to do here, just set top to NULL
2. Copy ctor
   1. Used assignment operator to assign one object to the current object.
3. Dtor
   1. Just used clear function
4. Assignment operator
   1. First I cleared the current object. Then I assigned the current top value to the rhs top value. I created temporary nodes and just incremented through the nodes and copied each one until the indexnode hit the end.
5. Push
   1. Didn’t really see the necessity of the logic\_error exception, but I put everything in a try catch block. I created a new pointer node with the appropriate data and point top to the node.
6. Pop
   1. First checked to make sure the linked list wasn’t empty, then created a temporary node to point to the top node. I deleted this node then return the data of the deleted node.
7. Clear
   1. I ran a loop until the Linked List was empty and popped each value
8. isEmpty
   1. if the top node was NULL it would be empty, otherwise there would be a value
9. isFull
   1. always set to false
10. showStructure
    1. first made sure the linked list wasn’t empty, then ran through the list and printed out every dataItem from each node.

Commands:

H : Help (displays this message)

+x : Push x

- : Pop

C : Clear

E : Empty stack?

F : Full stack?

Q : Quit the test program

list is empty

Command: +x

Push x

[ x ]

Command: -

Popped x

list is empty

Command: +x

Push x

[ x ]

Command: +c

Push c

[ c x ]

Command: C

Clear the stack

list is empty

Command: E

Stack is empty

list is empty

Command: F

Stack is NOT full

list is empty

Command: Q

PostFix.cpp

I included <cstdio> but I’m pretty sure <iostream> provides getchar() functionality as well. Otherwise, I just kept reading in the chars until it hit a new line. I used getchar because cin would skip over the newline operator and cause errors. Then once in the loop, I continued the loop if it read a space, and if it didn’t I separated the read in character as either an operand or an operator. If it was the operator, I would pop 2 values off the stack and apply the operator to them.

1. 4 3 + 2 3 ^ /
   1. (4 + 3 ) / 2^3
   2. Enter the Postfix expression: 4 3 + 2 3 ^ / result: 0.875
2. 4 3 \* 2 3 - +
   1. 4 \* 3 + (2 - 3)
   2. Enter the Postfix expression: 4 3 \* 2 3 - + result: 11
3. 9 3 3 ^ /
   1. 9 / 3 ^ 3
   2. Enter the Postfix expression: 9 3 3 ^ /

result: 0.333333

1. 6 3 + 4 2 - ^
   1. (6 + 3) ^ (4 - 2)
   2. Enter the Postfix expression: 6 3 + 4 2 - ^

result: 81

1. 5 3 2 ^ \*
   1. 5 \* 3 ^ 2
   2. Enter the Postfix expression: 5 3 2 ^ \*

result: 45

Delimiters.cpp

I went through the expression one character at a time. If it wasn’t a ‘[‘ or ‘]’ it was ignored. Otherwise, I stored the ‘[‘ onto a stack and checked for a corresponding bracket if it was ‘]’. If the entire string is read through, I check that the stack is empty to ensure all brackets were corresponded to.

This program checks for properly matched delimiters.

Enter delimited expression (<EOF> to quit) :

[5 + 4] + 4]

Invalid

Enter delimited expression (<EOF> to quit) :

[[5 + 4] + 4 ]

Valid

Enter delimited expression (<EOF> to quit) :

[[5 + 4]

Invalid

Enter delimited expression (<EOF> to quit) :

[] + 4 [ 2 + 3]

Valid

Enter delimited expression (<EOF> to quit) :

[[[[]]]]

Valid

Enter delimited expression (<EOF> to quit) :

<EOF>