

Homework 6: STL Stacks and Queues

cs244

Due: November 4, 8:00 AM

Allowances, if any, for late work and related penalties will be as described in the syllabus. As always, check with the instructor for any amendments or changes.

Overview

Use STL stacks and queues to solve a set of problems.

Topics: STL, C++ classes, stacks, queues, algorithm design and implementation, code testing, peer reviews

Preparation

Step 1: Create a Folder

In Linux in your .../Documents/Programs folder

Create a folder named **A06**

This is necessary as you will compress the A06 folder and its contents for submission.

Step 2: Check D2L

There may be starter code for this assignment available on D2L.

Files to Create

The files for this assignment **MUST** be named:

saq.h saq.cpp saqMain.cpp README.txt

Your *main* function is to be located in saqMain.cpp.

All code files should have comments with the file name, your name, the last modified date, and a description of the file contents. Each function should be appropriately commented in the header and implementation files – most should include pre and post conditions. The body of each function may need comments for the less than obvious parts of the code (if any).

The final submission of files must include a file named: **README.txt**

Providing instructions on how to compile/link your program, for example:

```
g++ file1.cpp file2.cpp
```

And also provides instructions on how to run your program, for example:

```
After compile type: ./a.out
```

makefile

This assignment does not *require* a makefile but you can provide one or for this assignment you may apply the following

To Compile at the Command Line Type:

```
g++ saq.cpp saqMain.cpp
```

To Run the executable created at the Command Line Type: `./a.out`

Assignment Description

This assignment is derived from content found in Data Structures Using C++ 2nd Edition by D.S. Malik. Understanding the content of chapters 7 and 8 of that book as well as an examination of the code contained therein will be helpful in completing this assignment.

Create a C++ **class** named **SAQ** (short for **S**tacks **A**nd **Q**ueues).

The SAQ class must have the following public member functions:

- **bool CheckExpressionBalance(std::string express)**
 - The input to this function is an arithmetic expression (sent as an std::string)
 - The expression should end with a semi-colon, but the function should still work if it does not (and it should ignore any extra characters after the first semi-colon found in the expression)
 - It returns true if the expression contains matching grouping symbols (i.e. the symbols are balanced), otherwise it returns false. Where grouping symbols are:
 - Parentheses ()
 - Braces { }
 - Square Brackets []
 - For example the arithmetic expressions $\{25 + (3 - 6) * 8\}$ and $7 + 8 * 2$ contain matching grouping symbols
 - However, the expression $5 + \{ (13+7) / 8 - 2 * 9$ does not contain matching grouping symbols. Nor does the expression $(a + [b * c] - d)$
 - This function (and/or its helper functions) must use the STL stack data type
- **bool CheckIsPalindrome(std::string checkMe)**
 - The input to this function is a line of text stored in the *checkMe* variable
 - Using the sent line of text, this function creates a copy of the text while it changes each uppercase letter to lowercase, removes all spaces, and places each letter both in a queue and onto a stack. The function then verifies whether the line of text is a palindrome (returning true if it is, otherwise returning false)
 - Recall: A palindrome is a set of letters or numbers that is the same whether read forward or backward
 - "Step on no pets" is a palindrome
 - "1221" is a palindrome
 - "A" is a palindrome
 - This function (and/or its helper functions) must use the STL stack and queue data types
 - Recall stacks are LIFO and queues are FIFO

The SAQ class may use/require other member helper functions, constructors, destructors, overloaded operators, copy constructors... to be written. But the above explicitly listed functions must exist as they will be directly called from the testing/grading program.

Homework 6: STL Stacks and Queues

cs244

Hints

The `convert.cpp`, `reverse.cpp`, `queueExample.cpp` and possibly other examples posted on D2L might be useful in understanding how to use STL stacks and queues. With regard to using STL queues the following may be useful.

To use STL queues include the queue header file (and you will need *using namespace std*):

```
#include <queue>
```

Assume T is any type or class (int, float, double, char, string...). Then to declare an empty queue name q :

```
queue<T> q;
```

So now we have the object q :

to see if q is empty try:

```
q.empty()
```

to get the number of items in q , try:

```
q.size()
```

to push a variable name t of type T onto the end of q , try:

```
q.push(t)
```

to pop the front element of q off, try:

```
q.pop()
```

to just get the front item of q (not remove it), try:

```
q.front()
```

Homework 6: STL Stacks and Queues

cs244

Tester File

Design your main function, in the saqMain.cpp file, to demonstrate and test the capabilities of your SAQ class.

Grading

100 points possible

Assuming it is turned in on time:

Your program will be peer reviewed and evaluated by members of your class.

Each will assign a grade to it from 0 to 100.

The instructor will also evaluate your work, scoring it from 0 to 100.

The final grade will then be assigned by the instructor based on the three scores.

Grading will be based on compilability, runability, correctness, comments, programming style, and similar attributes. Further details may be given in class.

Turn-In Directions

Correctly submitting your work is worth 0 points, BUT if not done correctly will likely result in nothing to grade.

Preparation

In Ubuntu Linux browse to your A06 folder

Make sure your source code files are in the folder

**Right click on the A06 folder,
Select Compress...**

Set the file name to be A06_ *yourlastname*.tar.gz
where *yourlastname* is your last name

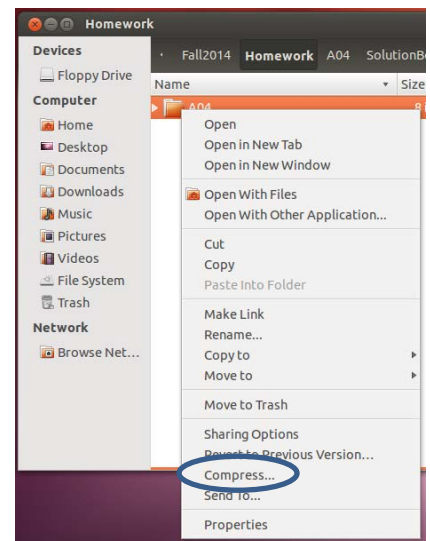
Example: if your last name is Gollygee
then the filename would be
A06_Gollygee.tar.gz

Click on the Create Button

This should create the file named A06_ *yourlastname*.tar.gz

Submit

Submit the A06_ *yourlastname*.tar.gz file
to the correct course drop box in D2L



Homework 6: STL Stacks and Queues

cs244

Peer Evaluations

For peer review evaluations you should follow the instructions given in class.

In most cases this will involve giving the above compressed files to the peers assigned to evaluate your program.

Again in most cases, they will then review and evaluate your program.

This often means your code will need to function with a test program they created.

This emphasizes the importance of coding to specification and doing so in a readable and maintainable manner.