Thursday, 08-12-2014

SWE315 : C++

Homework 17 (Real sequence is #10) - (Due date 08-19 : Doing most of it in class!!)

General notes:

1. Please send solution to: zbaharav@cogswell.edu
2. You know the drill by now: Simply hit reply, and no zipped directories etc..
3. Please attach this document with your solution (including screen shots!) at the bottom.

====

Assignment:
(This assignment is loosely based on a Queue implementation in: "C++ primer plus", by Stephen Prata, Chapter 12).

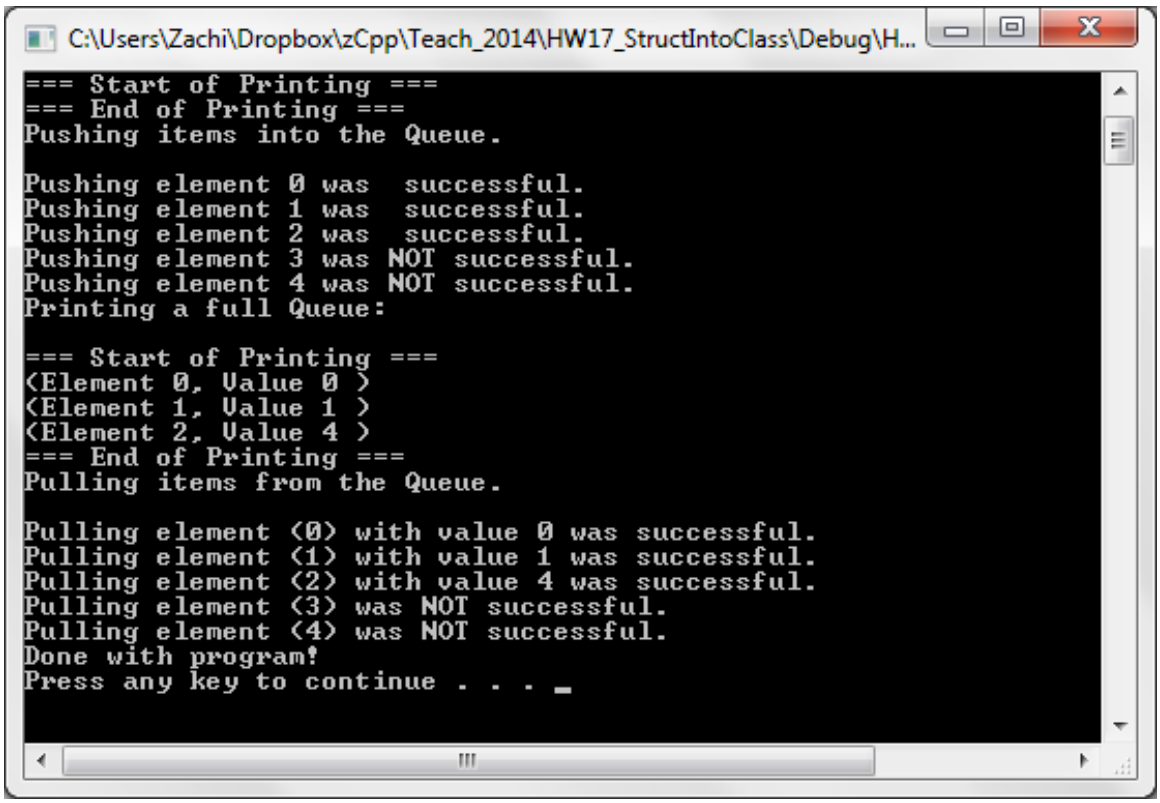
We wrote together a simple C++ implementation of a Queue. No classes were harmed in the process.
Your homework is to convert this into a proper C++ class implementation.

This is a continuation of our 'container class' exercises, though this time we are not looking at templates, but rather the class structure.

Here are some guidelines:

1. Define a class Queue, with the proper variables and interfaces.
2. In main, create a Queue object, and simply replace all the function calls with the appropriate class-method calls.
3. Do not use NodeP anymore (was just easier for our cumbersome implementation. Not needed in the real thing).
4. Hint: In your class it makes sense to have the qSize as a private variable const int. HOWEVER, be aware that you can initialize a const variables only with the initialization list.
5. Please leave the Node structure as is. This is NOT the goal of this exercise.
6. Please note: The newly created Class implementation functions need at most one argument.

Below is a screen shot of the program (which should look like the one you've got from the basic one in class…):



========

For your reference, below is the code we wrote in class:

/* File main.cpp */

```
#include <iostream>

using std::cout;

struct Node
{
	int data;
	struct Node* next;
};

typedef Node* NodeP;

bool isfull(const int items, const int qsize)
{
	return items==qsize;
}

bool isempty(const int items)
{
	return items==0;
}
```

```
bool AddToQueue(int& items, NodeP& rear, NodeP& front, const int qsize, const int
newdata)
{
	if (isfull(items, qsize))
		return false;
	Node* add = new Node;
	if (add==NULL)			// Just checking if you can create the
new object.
		return false;

	add->data = newdata;
	add->next = NULL;
	if (isempty(items))
		front = add;  // Place item at front of list
	else
		rear->next = add;
	rear = add;
	items++;
	return true;
}

bool RemoveFromQueue(int& dataReturned, int& items, NodeP& rear, NodeP& front,
const int qsize)
{
	if (isempty(items))
		return false;
	dataReturned = front->data;
	items--;
	//Node* tmp = front;
	front = front->next;
	//delete tmp;
	if (items==0)
		rear = NULL;
	return true;
}

void printQueue(const NodeP front, const NodeP rear)
{
	NodeP iterator;
	int num=0;
	iterator = front;

	cout << "=== Start of Printing ===\n";
	while (iterator != NULL)
	{
		cout << "(Element " << num++ << ", Value "<< iterator->data << "
)\n";
		iterator = iterator->next;
	}
	cout << "=== End of Printing ===\n";
}

int main(void)
{

	cout << "Starting Queue simulation.\n" ;

	NodeP front = NULL;  // First item in Queue
```

```
	NodeP rear = NULL;          // Last item in Queue
	int  items = 0;             // Items in Queue
	const int qsize = 3; // Maximum number of elements in Queue

	cout << "Printing an empty Queue:\n\n";
	printQueue(front, rear);

	cout << "Pushing items into the Queue.\n\n" ;
	for (int ii=0; ii <= qsize+1; ++ii)
	{
		bool b = ( AddToQueue(items, rear, front, qsize, ii*ii));
		cout << "Pushing element " << ii << " was " <<  (b ? " ": "NOT ") <<
"successful.\n"   ;
	}
	cout << "Printing a full Queue:\n\n";
	printQueue(front, rear);

	cout << "Pulling items from the Queue.\n\n" ;
	int val;
	for (int ii=0; ii <= qsize+1; ++ii)
		if (RemoveFromQueue(val,items, rear, front, qsize))
			cout << "Pulling element (" << ii << ") with value " << val <<
" was successful.\n" ;
		else
				cout << "Pulling element (" << ii << ") was NOT
successful.\n" ;


	cout << "Done with program!\n" ;
	system("pause");
	return 0;
}
```

/* End of File main.cpp */

=== End of Homework 17 ====

=== Please attach your solution below here, including screenshots ====