COMP2404A (Winter 23) -"Introduction to Software Engineering"

Tutorial 6 of 10 - Due Saturday, Mar. 11, 11:59 pm

- No late tutorials will be accepted.
- If there are specific instructions for making a function, please follow them exactly. That means that
  - function names
  - function return types
  - parameter types and order

  should all be **EXACTLY** as described. If the script can't read it, you will receive 0 for that part.
- Your **Tutorial 6** code will be marked by a Python script running on the Gradescope server. You are being given a similar script so that you may make sure your code runs correctly.

---

1 Submission Instructions

Download "tutorial6.zip". Unzip it into your working directory. There is a directory "tutorial6" and the test file "t6test.py". In the "tutorial6" folder are "Location.cc", "Location.h", "test.cc" and "defs.h" to get you started. To the "tutorial6" folder you should add the following files.

1. Header and source files for the `Order` class described below.
2. Header and source files for the `Queue` class described below.

Once you have written the classes and completed your tests, submit this tutorial to Gradescope. The Gradescope server is flexible in how you submit. You may zip the folder, zip the files, or submit the folder itself or individual files. What follows are one set of instructions that will work on Gradescope. You will zip the "tutorial6" directory into a file "tutorial6.zip". If you are doing this in the course VM you must do this from the command line. Open a terminal in the folder that contains "tutorial6". Use the command `zip -r tutorial6.zip tutorial6`. This will zip the `tutorial4` folder, or update it if you change the contents. DO NOT USE tar FILES. These do not seem to work well with Gradescope. Submit to Gradescope by the deadline. You will receive your mark immediately, and you may submit as many times as you like. You may also submit tutorials up to one week late for a 10% penalty.

2 Testing Your Tutorial With `t6test.py`

`t6test.py` is a test script that is very similar to the script that is being run on Gradescope (there might be slightly different input or different even tests). So the mark you see here should be close to the mark you will receive on Gradescope. To run `t6test.py`, open a command line in the directory containing `t6test.py`. You may have to make it executable, so type `chmod +x t6test.py`. You may run the script as is, in which case it will look for a file to unzip. Or, if you have not zipped your files yet you may supply a "-nozip" argument, in which case it looks for the "tutorial6" folder.

To have the script unzip `tutorial6.zip` and then test your code, run `./t6test.py`. To skip the unzip step use `./t5test.py -nozip`. When your tutorial is being officially marked you SHOULD NOT SUBMIT A .tar FILE. Either submit a zipped file or drag and drop the tutorial6 folder into Gradescope.

Running this script will generate a file "results.txt" just outside of the "tutorial6" folder. This will have some useful output as well as the mark.

3 Learning Outcomes

In this tutorial you will learn how to make the `Queue` class, which is a cousin to the `LinkedList` class.

4 Instructions

4.1 Overview

In this tutorial you will complete two classes from Assignment 3: the `Order` class (Section 5.3) and the `Queue` class (Section 5.4). There is one test file provided, `test.cc` that will test the functions that you provide from those classes. You are provided with a Makefile - make any changes you see fit (and remember that Linux is case sensitive!).

4.2 Order Class

Complete Section 5.3 in Assignment 3. Note that the `Order::print` function makes use of the `Franchise::menu` static variable. For the purposes of this tutorial you may skip that part and simply print out the `menuItem` number (or you could include the `Franchise` class, or use any other solution you like) in your `Order::print` function.

4.3 Queue Class

Complete Section 5.4 in Assignment 3.

4.4 Makefile

Your Makefile should compile two object files, `Student.o` and `Queue.o`. It should link these object files to the `test` executable. In addition your Makefile should contain an `all` command that creates the `test` executable and a `clean` command that removes all executables and object files.

4.5 t6test.py

Run this python script to test the classes described above. Correct all errors.