COSC1076 | Advanced Programming Techniques

Tutorial/Lab | Week 08

Overview

The Week 08 tutorial/lab is to revise content from the last few weeks.

- Linked Lists
- Methodologies of Programming

Tutorial Questions

The tutorial questions are limited to allow more time for talking with your tutor about your assignment 2.

Consider the following questions:

1. Briefly discuss the purpose of the programming methodologies of:
   (a) Structured Programming
   (b) Defensive Programming
   (c) Programming by Contract
2. For these consider questions such as:
   (a) Why do they achieve?
   (b) When is it appropriate to use each methodology?
   (c) Why would you not use the methodology?

Assignment 2 Progress Update

Update your tutor on the progress of your group. You might consider such things like:

- ***What has each individual done in the past weeks?***
- ***What is the goal of each individual in the group for the coming week?***
- Do you have a design of the classes, data structures, and functions for the assignment?
- Have you got any of the basic functionality working?
- Have you contemplated the final state of the program, such as the enhancements your group may complete?
- How is your group managing their codebase?
- How will your group co-ordinate/discuss issues?

Week 08 Quiz

Remember that at some point this week you need to complete the online Canvas Quiz this week. You only have:

- 2 hours to complete the quiz.
- 1 attempt at the quiz, so make sure you set yourself enough time.

Do not attempt the quiz in your lab. You will need to talk with your tutor and go through the tutorial questions. Further, your tutor *will not* help you with the quiz. If you run out of time in the lab, you will not be able to start the quiz again.

Lab Questions

It is a good idea to attempt the lab questions before coming to class. The lab might also be longer than you can complete in 2 hours. It is a good to finish the lab at home.

You should demonstrate your work to your tutor.

Exercises

1. If you are unfamiliar with using Git, then work through the very well put together GIT Tutorials from Atlassian[1]. Only complete the sections:
   (a) Beginner
   (b) Getting Started
   (c) Collaborating

2. The below ADT defines a Linked List, as was used in Lectures and the Week 07 Lab. It also includes two additional methods for removing elements from the Linked List.

Node.h

```
class Node {
public:

   Node(int data, Node* next);
   Node(Node& other);

   int    data;
   Node*  next;
};
```

LinkedList.h

```
#include "Node.h"

class LinkedList {
public:
   LinkedList();
   ~LinkedList();

   int size();
   void clear();
   int get(int i);

   void addFront(int data);
   void addBack(int data);

   void deleteFront();
   void deleteBack();

private:
   Node* head;
};
```

   Implement the methods:
   (a) `deleteFront`
   (b) `deleteBack`

3. Now consider different implementations of the `deleteFront` method under the *programming by contract* paradigm. The `deleteFront` method has a potential issue if the linked list is empty (that is, the head is

[1] https://www.atlassian.com/git/tutorials

null). Implement `deleteFront` for each of the following contracts:

(a) `deleteFront` may be called at any time. If the list is empty the method returns without modifying the list, and without generating an error.

(b) `deleteFront` may only be called if the list contains at least ONE element. If `deleteFront` is called on an empty list, the behaviour is undefined.

(c) `deleteFront` may only be called if the list contains at least ONE element. If `deleteFront` is called on an empty list, the method throws an `std::runtime_error` exception.