COSC1076 | Semester 1 | Advanced Programming Techniques

Tutorial/Lab | Week 07

Overview

The Week 07 tutorial/lab is to revise content from the last few weeks.

- Linked Lists
- Methodologies of Programming

Structure of the Lab

There are four tasks to complete for this week's lab.

1. Forming groups for Assignment 2, and informing your tutor.
2. Lab on using Vectors and Linked Lists.

We suggest that you:

1. Spend the first 10 minutes making sure everybody is in a group.
2. Setup your group collaboration tools including:
   (a) Github/Bitbucket project
   (b) MS Team
3. Add your tutor to your group collaboration tools
4. Then move onto the lab questions.

Group Formation for Assignment 2

Assignment 2 will be conducted in groups of 4.

1. You *may* form groups with any student in the course, however
2. **We STRONGLY recommend** that you form groups from within your labs, because:
   (a) It is ultimately *your* responsibility to make sure you form a group. Assignment 2 **is not** an individual assignment, thus there are significant penalties for failing to form a group in a timely manner.
   (b) Your tutor will help you form groups, *but only within your lab.*
   (c) You will have plenty of opportunity to discuss your group's progress and get help from your tutor during the rest of the course. It will be *extremely helpful* for your *whole group* to be present, but this can't happen if you have group members outside the lab

Groups for Assignment 2 must be ***confirmed*** and ***registered*** with your tutor by ***your week 8 lab***. Your tutor "register" your group by adding the group onto Canvas.[1]

Setup your Group Git

For Assignment 2, we will require you to create and use a Git Repository with your group members. In this lab you will need to:

- Create a new *private* Git repository on BitBucket or GitHub. Your repository ***must*** be private. It is important that your work is not publicly accessible.

  > ! It is ***academic misconduct*** to make a public repository of your assignment work. If you make your work public and another student copies it, *you are also responsible and may receive a penalty.*

- Add your tutor to your repository. Your tutor will provide their details *in your lab.* These details may not be posted publicly, so ensure you attend your lab. If your group members are in different labs, add **all** of your tutors to the repository. (You may need to wait for your lab in Week 7 to receive these details. Please do not spam tutors with emails asking for their details which they will be provided in class.)
- Clone the new repository to your local computer.
- Download the example code files for this week's lab and add them to the new repository
- Make an initial commit with these text files and push the commit to BitBucket. We recommend using Sourcetree to help you do this.

[1] If your group spans multiple labs, have one of your tutors register the group.

Setup your Group MS Team

For Assignment 2, we will require your group to use **MS Teams** for groups discussions and managing your group work. This is because:

1. MS Teams is linked to your student login credentials
2. On external tools we are unable to verify the identity of users
3. In the event of group issues, we will be able to view the entirety of all discussions

Please note that if required you may need to provide evidence of your group work and contributions. We will only accept as evidence information that is recorded on RMIT tools and systems. Thus we will not accept records from platforms such as Slack and Discord, for the reasons listed above. In this lab you will need to:

- Create a new RMIT MS Team group.
- Add your tutor to your MS Team. If your group members are in different labs, add **all** of your tutors to the repository.

Lab Questions

> This week's Lab will **be very helpful** for working on Assignment 2.
> For this reason, we recommend that you **work on the lab questions with your group members**. You can use collaborate ultra break-out rooms to do this. You could also use VSCode Live to help share your code and work on it together!

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.

The code snippets can be accessed from the COSC1076 Git Repository.

Exercises

1. If time permits, discuss with your tutor the following programming methodologies of:
   (a) Structured Programming
   (b) Defensive Programming
   (c) Programming by Contract
   For each of 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?

2. *(In your groups)* The below ADT defines a Linked List, as was used in Lectures and the Week 06 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 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 exception.