COSC1076 | Advanced Programming Techniques

Tutorial/Lab | Week 09

Overview

The Week 09 tutorial/lab is to revise content from the last few weeks.

- Linked Lists
- Smart Pointers
- Scope

Tutorial Questions

The tutorial questions relate to the following C++ program

```
 1 #include <iostream>
 2 #include <memory>
 3 
 4 #define EXIT_SUCCESS    0
 5 
 6 using std::cout;
 7 using std::endl;
 8 using std::shared_ptr;
 9 
10 void foo(shared_ptr<int> arg);
11 
12 int main(void) {
13    shared_ptr<int> x = std::make_shared<int>(0);
14    shared_ptr<int> y = std::make_shared<int>(-10);
15    cout << "x: " << *x << ", y: " << *y << endl;
16 
17    foo(x);
18    cout << "x: " << *x << ", y: " << *y << endl;
19 
20    y = x;
21    cout << "x: " << *x << ", y: " << *y << endl;
22 
23    {
24       int x = 2;
25       shared_ptr<int> y = std::make_shared<int>(7);
26       cout << "x: " << x << ", y: " << *y << endl;
27    }
28 
29    cout << "x: " << *x << ", y: " << *y << endl;
30 
31    return EXIT_SUCCESS;
32 }
33 
34 void foo(shared_ptr<int> arg) {
35    *arg += 1;
36 }
```

1. What is the output of the program?
2. For each shared pointer, draw or describe how the value of each variable is changed.
3. For each shared pointer, discuss where the associated memory is allocated and de-allocated
4. Describe what happens to the variable `x` in the block at lines 22-27.

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?
- Have you written any tests?

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. The below ADT defines a Linked List, as we have been using in the past few weeks.

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();

   void addAt(int i);
   void deleteAt(int i);

private:
   Node* head;
};
```

Implement the methods:
(a) `addAt(int i)`
(b) `deleteAt(int i)`

2. Change the LinkedList and Node classes to use C++ shared pointers, as described in the header files below, and re-implement the methods of the Linked List class.

Node.h

```cpp
#include <memory>
class Node {
public:

   Node(int data, std::shared_ptr<int> next);
   Node(Node& other);

   int data;
   std::shared_ptr<int> next;
};
```

LinkedList.h

```cpp
#include "Node.h"
#include <memory>

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();

   void addAt(int i);
   void deleteAt(int i);

private:
   std::shared_ptr<int> head;
};
```