Discipline of Computing and Information Technology
Semester 2, 2021 - SENG1120

Assignment 2

Due using the Blackboard Assignment submission facility:
11:59PM – October 10th, 2021 (end of Week 10)

NOTE: *The important information about submission and code specifics at the end of this assignment specification.*

INTRODUCTION

In lectures we have discussed the use of **templates** to provide the compiler with blueprints for functions and classes. These are used by the compiler to produce code that implements functions and/or classes that are suitable for the type(s) to which you apply them in your program code.

PROBLEM DESCRIPTION

Your task in this Assignment is to modify your Assignment 1 code in the following ways:

- Replace the `Node` and `Linkedlist` classes by **class templates**. Therefore, when an instance of `Node` is created, the syntax will be `Node<EToll>`. That allows for a more generic code, eliminating the need for `typedef`.
- Use an `LStack` class template as a wrapper class around `LinkedList`. `LStack` will use a `LinkedList` instance as a member variable. When instances of `EToll` are added to `LStack`, they will be stored in the member `LinkedList` using methods from `LinkedList`. Note that the implementation of `LStack` must comply with the public interface and logic of stacks (elements are added and removed from the top of the stack, and internal nodes are not visible). Moreover, removing instances of `EToll` from the stack, and printing it **can only use methods from `LStack`** (i.e. `push(i)`, `pop()`, `peek()`, `isEmpty()`). That is, if you want to remove an item from the stack, you can't simply call the method `remove` from `LinkedList`, as it would violate the logic of stacks. **This constraint will help students understand the dynamics of stacks, and the fact that only the top (*first*) element is visible at any time.**

For SENG1120 students who want to be challenged more, there is an extra requirement, worth a bonus 2.0 marks:

- Create a class `TollStack` that extends `LStack` via **inheritance**. `TollStack` will declare and implement all the methods that are specific to the toll application, leaving all the native functionality of a generic stack in `LStack`.
- The only methods allowed in the `LStack` class are the ones that would not change if the data type changed (`push(i)`, `pop()`, `peek()`, `isEmpty()`, `+=`). All other methods that require a specific implementation for the `EToll` application (`remove()`, `count()`, `totalIncome()`, `<<`, `-=`) should be declared and implemented in the `TollStack` class.

NOTES ON MAKEFILES

Two makefiles have been provided – depending on whether or not you are attempting the bonus task – therefore you will have to rename one of these to simply `makefile` in order to compile your program correctly.

SUBMISSION NOTES

Make sure your code works with the files supplied, and DO NOT change them. For marking, we will add the demo file to the project and compile everything using the makefile, together with your own files. If it does not compile or run, your mark will be zero.

Your submission should be made using the Assignments section of the course Blackboard site. **Incorrectly submitted assignments will not be marked.** You should provide the `.h` and `.cpp/.hpp` files related to the `LStack`, `LinkedList` and `Node` classes (plus `TollStack` if you are doing the Bonus Task), and your selected `makefile` only, plus an assessment item coversheet. Also, if necessary, provide a `readme.txt` file containing instructions or comments for the marker. Each program file should have a proper header comment section including your name, course and student number; and your code should be properly documented.

Remember that your code should compile and run correctly using Cygwin. There should be no segmentation faults or memory leaks during or after the execution of the program.

Compress all your files into a *single .zip file*, using your **student number** as the filename. For example, if your student number is **`c9876543`**, you would name your submission:

`c9876543.zip`

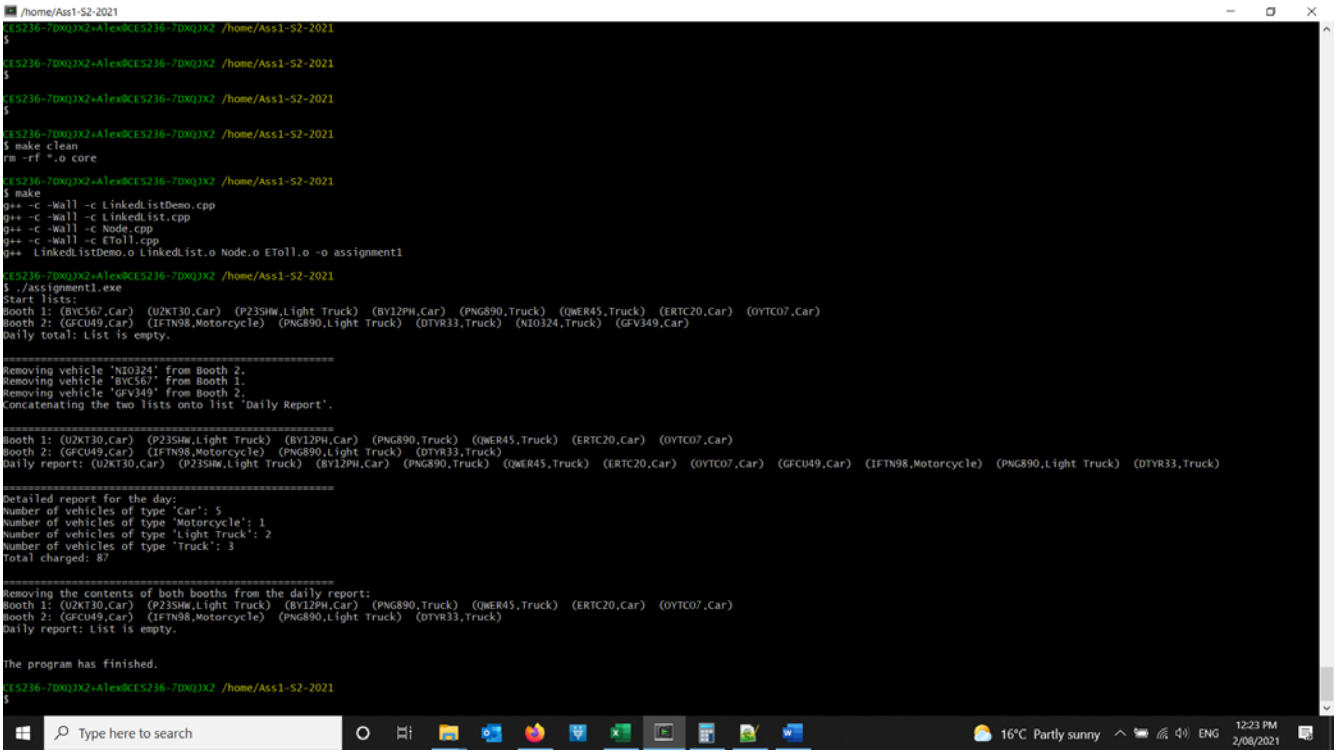
If you have attempted the Bonus Requirement (*or you are a 6120 student*), please include a blank text file in the same folder as your source files, simply called **`Bonus.txt`** – this is to make it clear to the marker that you are attempting this.

Submit by selecting the **Assignment 1** link that will be found in the **Assessment** section on **Blackboard**.

DO NOT USE RAR or any compression format other than ZIP. Late submissions are subject to the rules specified in the Course Outline. Finally, a completed **Assignment Cover Sheet** should accompany your submission, as well as any Adverse Circumstances documentation, should you have applied for and been granted Adverse Circumstances before your submission!

This assignment is worth 10 marks of your final result for the course.

Compiling and running your files together with the demo files provided should output the same result from Assignment 1. *Note that in the demo files provided, the element (PNG890, Light Truck) was replaced by an element (UTW951, Light Truck), as it had the same licence as another vehicle and might cause confusion.*



```
CES236-7DXQJX2+Alex@CES236-7DXQJX2 /home/Ass1-S2-2021
$ make clean
rm -rf *.o core

CES236-7DXQJX2+Alex@CES236-7DXQJX2 /home/Ass1-S2-2021
$ make
g++ -c -Wall -c LinkedListDemo.cpp
g++ -c -Wall -c LinkedList.cpp
g++ -c -Wall -c Node.cpp
g++ -c -Wall -c EToll.cpp
g++  LinkedListDemo.o LinkedList.o Node.o EToll.o -o assignment1

CES236-7DXQJX2+Alex@CES236-7DXQJX2 /home/Ass1-S2-2021
$ ./assignment1.exe
Start lists:
Booth 1: (BYC567,Car)  (U2KT30,Car)  (P23SHW,Light Truck)  (BY12PH,Car)  (PNG890,Truck)  (QWER45,Truck)
(ERTC20,Car)  (OYTCO7,Car)
Booth 2: (GFCU49,Car)  (IFTN98,Motorcycle)  (PNG890,Light Truck)  (DTYR33,Truck)  (NIO324,Truck)  (GFV349,Car)
Daily total: List is empty.

=====================================================
Removing vehicle 'NIO324' from Booth 2.
Removing vehicle 'BYC567' from Booth 1.
Removing vehicle 'GFV349' from Booth 2.
Concatenating the two lists onto list 'Daily Report'.

=====================================================
Booth 1: (U2KT30,Car)  (P23SHW,Light Truck)  (BY12PH,Car)  (PNG890,Truck)  (QWER45,Truck)  (ERTC20,Car)
(OYTCO7,Car)
Booth 2: (GFCU49,Car)  (IFTN98,Motorcycle)  (PNG890,Light Truck)  (DTYR33,Truck)
Daily report: (U2KT30,Car)  (P23SHW,Light Truck)  (BY12PH,Car)  (PNG890,Truck)  (QWER45,Truck)  (ERTC20,Car)
(OYTCO7,Car)  (GFCU49,Car)  (IFTN98,Motorcycle)  (PNG890,Light Truck)  (DTYR33,Truck)

=====================================================
Detailed report for the day:
Number of vehicles of type 'Car': 5
Number of vehicles of type 'Motorcycle': 1
Number of vehicles of type 'Light Truck': 2
Number of vehicles of type 'Truck': 3
Total charged: 87

=====================================================
Removing the contents of both booths from the daily report:
Booth 1: (U2KT30,Car)  (P23SHW,Light Truck)  (BY12PH,Car)  (PNG890,Truck)  (QWER45,Truck)  (ERTC20,Car)
(OYTCO7,Car)
Booth 2: (GFCU49,Car)  (IFTN98,Motorcycle)  (PNG890,Light Truck)  (DTYR33,Truck)
Daily report: List is empty.


The program has finished.

CES236-7DXQJX2+Alex@CES236-7DXQJX2 /home/Ass1-S2-2021
$
```

Dan and Alex

v1.0 2021-09-10