CSCI204/MCS9204/CSCI804
Object and Generic Programming in C++
Laboratory Exercise 4 (Week 5)

Task: Stack and linked list (1.0)

In this task you will create a singly linked list with classes and manipulate the linked list.
Define a class **Node** in a file **mylist.h** with two data members: one is an integer and the other one is a pointer of the Node. Define the necessary member functions to set and get values for a Node instance.
Implement the member functions of class Node in a file mylist.cpp.

Define a class **MyList** in the file **mylist.h** with private data member **head -** a pointer points to the head of a linked list. Define the following member functions:

- Default constructor.
- Destructor.
- push_front(int) function adds a new integer value at the front of the linked list.
- pop_front() function removes the node at the front of the linked list.
- front() function returns the value at the front of the linked list.
- isEmpty() function returns true if the linked list is empty.
- Other necessary member functions.

Implement the member functions of class **MyList** in the file **mylist.cpp**

Define a class **MyStack** in the file **mylist.h** by using MyList instance to simulate a stack. Define the following member functions for the class:

- Default constructor.
- isEmpty() function return true if the stack is empty.
- push(int) function pushes an integer value on the stack top.
- pop() function pops up the stack top element.
- stackTop() function returns the value at the top of the stack.

Implement the member functions for the class **MyStack** in the file **mylist.cpp**.

Implement driver program include main function in a file **testMain.cpp** to get input integers, push them onto a stack, then pop all the values from the stack and print out those values.

Note: throw an exception when pop data from an empty stack or linked list.

Testing
You can compile the program like

```
g++ testMain.cpp mylist.cpp
```

When we run the program, the results may look like the following (red data mean inputs from keyboard):

```
./a.out < input.txt
Input integers: 2 1 3 8 15 32 19 23
Pop from the stack: 23 19 32 15 8 3 1 2
```

You can download testing file “input.txt” from the Moodle site.

Submission:

You should submit the files of task One and Two to the server by 11:59 PM on Friday, 28 August 2015 via command:

```
submit –u your-user-name –c CSCI204 –a L4  mylist.h mylist.cpp testMain.cpp
```

and input your password.

Make sure that you use the correct file names. The UNIX system is case sensitive. You must submit all files in one *submit* command line.

After submit your assignment successfully, please check your email of confirmation. You should keep this email for the reference.

You would receive ZERO of the marks if your program codes could not be compiled correctly.

Later submission will not be accepted. Submission via e-mail is NOT acceptable.

End of Specification