**Lab 5: Stacks using Linked List**

**Objectives:**

1. Students will implement a stack data structure using a templated single-linked list class.

**Background:**

In this lab students will learn how to design a basic stack data structure using a single-linked list. A stack is just taking the underlying data structure and putting limits on where in the structure you can add/remove/access/change values. In the case of stacks, all of these operations are only done on one end of the underlying data structure. Students should take advantage of how each underlying data structure is implemented to design a stack such that the stack operations are efficient (O(1) instead of O(n) when possible).

**Directions:**

1. Open Visual Studio 2010 and select C++ Development Settings. Create a new **empty** C++ project for the lab called **Lab5**.
2. Download the files tester.cpp and stack\_list\_tpt.h from D2L into your project folder (**Documents/Visual Studio 2010/Projects/Lab5/Lab5**).
3. In the solution explorer in Visual Studio, right click the project (**Lab5**), click

Add->Existing Item. Select the 2 copied lab files and add them.

STACK USING LINKED LIST

1. Edit stack\_list\_tpt.h to implement the requested methods.

**NOTE**: Use T() to get datatype neutral value when needed.

**HINT**: Treat the **head of the list** as **the top of the stack**.

1. *Add the logic for the following methods:*

* push
* pop
* top

1. Compile and Run the test code and determine if there are any errors in the code. Correct any errors until the code compiles successfully and runs with expected output.

**Turn In:**

* Source code for the linked list stack class (stack\_list\_tpt.h)