CMPT 225 Lab 9: C++ Template Classes

Creating a Template

C++ allows a programmer to create **templates** that can be instantiated. A template allows objects to be created that can store (or use) data of any type.

In this lab you will convert the **int** linked list class to a template class. Start by downloading the zipfile. It contains:

- LinkedList.h
- LinkedList.cpp
- A makefile. **This makefile won't work unless you follow the instructions below.**
- template_test.cpp, a driver program you should fill in

To convert the class to a template class you must:

1. Put the entire class (the .h file class definition, and the .cpp file method implementations) in one .h file. For this lab, put the entire class into LinkedList.h. This means you should copy most of LinkedList.cpp (all of it except the **#include** directives) into the bottom of LinkedList.h.
2. Preface the **class definition** and **each member function implementation** with this line: **template <typename Type>**
3. The class qualifier (**LinkedList::**) that appears before a method name in each implementation, should be replaced by **LinkedList<Type>::**
4. Whenever the type that the class contains (**int** in our case) is referred to it should be replaced with the word **Type.** Note that you should not replace those occurences of **int** that do not refer to the data in a node or in the list.

For example here is the add method before and after these changes:

```
void LinkedList::add(int x){
        Node *p = new Node(x);
        // Assign appropriate values to the new node
        p -> next = head;

        // Make the head point to the new node
        head = p;
   size++;
}
```

and the "templated" version

```
template <typename Type>
void LinkedList<Type>::add(Type x){
        Node *p = new Node(x); //temporary node
        // Assign appropriate values to the new node
        p -> data = x;
```

```
        // Make the head point to the new node
        head = p;
    size++;
}
```

To use your linked list template class in a program you need to specify what type is to be stored:

- **LinkedList<int>** creates a linked list of ints, and
- **LinkedList<string>** creates a linked list of strings

Testing

Test your linked list template by modifying the driver program in template_test.cpp so that it creates an **int** linked list and a **string** linked list, and uses each of the LinkedList methods at least once for each type of list. (Note that the provided version of template_test does **not** do this for the LinkedList with int hardwired in.)

When you have done this, please show a TA your modified driver program and the results of running it to receive your marks for this lab.