University of Wollongong

School of Computing and Information Systems

CSCI251/851 **Advanced Programming** **Autumn 2018**

Exercise 3 **(Due: 11.59pm Week 8, Friday 27 April)** **3 marks**

Aim:

This exercise is to familiarise you with the implementation and use of dynamic container classes with overloaded operators.

Requirements:

The LinkedList class given below is to be used to implement a stack. Meaning, items are to be added and removed from the head only. Complete the implementation of the given LinkedList stack class by following the steps below. After completing each step, test that your LinkedList class works and behaves like a stack by uncomenting the appropriate code in main.cpp and checking the output. Each step is worth 1 mark each.

Step 1

Implement all the public member functions of the LinkedList class declared below in files list.h and list.cpp. Test your code by uncomenting the "Step-1" code in main.cpp.

```
struct Node
{
    int Item;
    Node *Next;
};

class LinkedList
{
    public:
        LinkedList();
        ~ LinkedList();
        void AddHead(int Item); // adds item to head of linked list
        int RemoveHead();       // removes item from head of list
        bool IsEmpty();         // returns true if list is empty
        void Print();           // prints list. eg 12 34 21 26
    private:
        Node *Head;
};
```

Step 2

Implement a copy constructor that makes a deep copy of the LinkedList argument. Uncomment the code in main() to test that the copy constructor works on both on empty and non-empty lists. Make sure the contents of the lists (stacks) are printed as expected and there are no memory leaks.

Step 3

Implement an assignment operator in your LinkedList class. The assignment operator should ensure that multiple assignments are possible. E.g.:

```
A = B = C;   // assign (copy) stack C to A and B
```

Test the assignment operator by uncomenting the "Step-3" code in main.cpp.

Submit:

Submit your files using the submit facility on UNIX as shown below:

```
$ submit -u login -c CSCI251 –a ex3 main.cpp list.h list.cpp
```

where 'login' is your UNIX login ID
Note: CSCI851 should also submit to –c CSCI251.

You must also demonstrate your program in your week 8 lab class. Failure to demo on time, without being granted an extension, will result in a 1 mark deduction for each week late. Late submissions without granted extension will receive a deduction of 0.5 marks for each day late.