CS 2420 Section 2 and Section 3
Spring 2016
Assignment 2: Linked List
Due: 11:59 p.m. February 9 (Tuesday), 2016
Total Points: 100 points

In this assignment, you will implement set operations. For detailed discussion on sets, read the examples at http://www.wiu.edu/users/mfmk/Math101/Sets/SOper.html. One of the efficient solutions is to use the ordered linked list. You will implement a Set class using the linked list class that was given in lecture. Each Set class instance will hold a set of **unique** integer values ranging from 1 to 99. You are to overload the following selected operators in the Set class.

- [10 points] < : proper subset – returns true if S1 is a proper subset of S2 (S1 and S2 cannot be the same sets).
- [10 points] ^ : intersection – returns a new set that is the intersection of S1 and S2. The intersection of two sets includes the elements from both sets without duplicates. For example, if S1={4, 8, 9} and S2={2, 3, 4, 7, 8, 9}, then S1^S2={4, 8, 9}. [Note: The contents in S1 and S2 remain unchanged after this operation!]
- [10 points] + : union – returns a new set that is the union of S1 and S2. Remember, there are no duplicates in the unioned set. For example, for the same S1 and S2 in the previous example, S1 + S2 = {2, 3, 4, 7, 8, 9}. [Note: The contents in S1 and S2 remain unchanged after this operation!]
- [10 points] − : set difference – all elements that are elements of S1 but not of S2. For example, if S1={1, 4, 5, 8, 9} and S2={2, 3, 4, 7, 8, 9}, S1 – S2 = {1, 5}. [Note: The contents in S1 and S2 remain unchanged after this operation!]

In addition, your Set class should provide the following methods and features:

- [5 points] Insert: Add a qualified integer in the list in the ascending order. The Insert operation does not allow duplicates. If a duplicate integer is attempted for insertion, no action is taken.
- [5 points] Delete: Delete an integer from the list.
- [5 points] Find: Return true if the integer is found in the list. Return false if the integer is not found in the list.
- [5 points] Print: Print all the elements in the linked list.
- [5 points] Overloaded = operator: Make a copy of the list to another list
- [5 points] Overloaded [] operator: Return the element at the position specified. For example, Set[0] will return the first integer in the set.
- [5 points] Copy constructor and constructor.
- [5 point] Destructor.

[10 points] After the Set class is implemented, you must write a driver program (main program) to test all the operations and demonstrate you have correctly implemented the requirements. The grader will create his own driver code to test all the functionalities.

Below is the Set.h file, which contains C++ Set class interface, for your reference.

```
#ifndef _Set_H
#define _Set_H

// Create a data type for the linked list node
class ListNode
{
public:
        ListNode();   // default constructor
        ListNode(int s);  // constructor with one parameter
        int value;
        ListNode *next;
};

class Set
{
private:
        ListNode *head;
        int count ; // the number of elements in the list

public:
        Set(); // Constructor: Create the initial linked list with two nodes
        Set(Set &obj);  // Copy constructor
        ~Set(); // Destructor

        //Functions
        void Insert(int v); // Insert a new value in order.  Duplicated element will not be inserted
        void Delete(int v); // Delete a value
        void Print();     // Display list in order
        bool Find(int v);  // Search v in the list. Return true if v is found; otherwise, return false

        // Overloaded operators
        bool operator<(const Set &obj); // proper subset
        Set  operator^(const Set &obj);  // intersection
        Set operator+(const Set &obj);  // union
        Set operator-(const Set &obj);  // set difference
        Set & operator=(const Set &obj); // assignment
        int operator[](const int index); // index
};

#endif
```

[10 points] Use the convention mentioned on slide 19 of Ch1.AlgorithmAnalysis.pdf to write the pseudocode for the intersection operation. Give the worst-case running time of this algorithm using the big-*O* notation.

Submission: Set.h, Set.cpp, main.cpp, and xxx.docx (xxx.txt or xxx.pdf)