CS 2433 C/C++ Programming

Programming Assignment 9
25 points
Due date and time: 11:59 PM, Monday, November 28, 2016 (This is the Monday following Thanksgiving break.)

Objective: Implement operator overloading and use a friend function.

Your submission must be comprised of two files:

`p9.cpp`	Contains the `main()` function and fully exercises the class `SetClass`.
`SetClass.h`	Contains the declaration and complete implementation of the class, Which implements the `abstract` data type `Set`.

You are required to design and implement a C++ class, named `SetClass`, to implement the abstract data type `Set`. The universal set is the integers 0 through 50 (inclusive). You must implement the set operations "union", "intersection", and "complement" as overloaded operators on operands of class `SetClass`. You MUST NOT implement these as member methods of the class. You must implement a function print as a friend of the class `SetClass`. Implement a `main()` procedure to test the class implementation. The requirements are given below. (Note: Penalty points exceed points possible, but no score can be lower than 0.)

1) (6 point penalty) All code required to fulfill parts 2, 3 and 4 of these requirements must be defined in a header file `SetClass.h`, while all code required to fulfill all other requirements must be in a file named `p9.cpp`, which must reference `SetClass.h` in a preprocessor `#include` statement. Both of these files must be submitted using the handin facility on the csx server.

2) (6 point penalty) Implement the following public member functions for `SetClass`:
   a. `isEmpty()` returns `true` if the `Set` is empty, `false` otherwise
   b. `isElement(x)` returns `true` if `x` is a member of the set, `false` otherwise
   c. `add(x)` adds `x` to the set. If `x` is already a member or out of range, return `false`, otherwise return `true`.
   d. `remove(x)` removes `x` from the set. If `x` was in the set (before removal), return `true` indicating the removal is successful, otherwise return `false`

   There should be no public members other than the ones listed above and constructors. The storage for `set` should be implemented as a protected member "`set`" (an array of `bool`, `bool set[51]`;). The constructors should initialize the `Set` object to be empty on creation. An element `i` is in the `set` if `set[i]` is true. In an empty set all elements of the array are `false`.

3) (6 points penalty) Implement a function "`print`" as a `friend` of the `class SetClass`. The function `print` takes an object of type `SetClass` and lists the elements of the `set`.

4) (4 points penalty) Implement the following functions on Sets (**these are not members of any class**):
   a. **union** as overloaded "+".
   b. **intersection** as overloaded "*".
   c. **complement** as overloaded "-".

5) (8 points penalty) Implement a main program that tests all operations by performing at least the following operations:
    a. Create three empty sets A, B, C (declare A, B, C); verify the sets are empty by printing their members
    b. Read a set of integers and build set A; read a second set of integers and build set B; print the sets A and B with proper identifications. You need to use the member function add to add elements to the set. If addition of any integer to the set returns false, issue an error message
    c. Read a set of numbers and remove them from B if it is present (you need to use the member function remove), issue appropriate message if any number is not in the set; print B
    d. Compute: C = A+B; print C
    e. Compute: C = A*B; print C
    f. Compute: C = -B, print C;
    g. Read an integer and check if it is in the sets A, B, and C. Issue appropriate message in each case. (i.e x is not in A)

6) (2 points penalty) All code must be well documented.

7) The program must compile and run on the csx server to get any credit.

CAUTION: This is an individual programming assignment. All work must be done individually. Sharing (or copying from any source) of code segment will be treated as plagiarism and dealt with accordingly. If any requirement is found to be ambiguous, bring it to the attention of the instructor for clarification.

Special note: Due to time constraints, it is not required that you actually save the integers that are added to the set, but can earn five (5) points extra credit if you do. Now, some explanation -- The Set container defined in C++ is just that, a container. It is intended to allow a program to insert and remove items from instances/objects of this class. This feature is essential for general purpose programming, as the container must correctly preserve the items inserted, since forcing programmers to implement separate processes for that would obviate the need for the defined Set container. In the present assignment, the bool array has the effect of preserving the items inserted indirectly by indicating presence via the state of the bool array's element accessed by the integer being added. This is possible due to the limitation of the universal set (in set notation either U or S can be used to specify the universal set) to a finite (and rather narrow) range of integers. The Set container itself must be able to support arbitrarily large number of items, and these may be composed of more than just an integer key-value.

Submit the solutions using the "handin" command. Example submission command is "`handin cs2433 program9 p9.cpp`", where `p9.cpp` is the file being submitted. Remember to submit the file `SetClass.h` as well, and to the same handin folder.