CS2400 HW7, Set Class (70 Points) Due: Thursday June 27, 2019

NO LATE SUBMISSIONS

Objective:

Design classes, arrays/vectors in classes, operator overloading, separate compilation.

You may use any function or library discussed in class or in the chapters we covered from your textbook. Do not use any other libraries or functions.

Design a class that stores a mathematical *set* of integers called ***MySet*** **(do not use a different name)**. You may assume that the set will never have more than 100 elements. The class should include the following functions:

- A default constructor that initializes a set to the empty set.
- Overload the "^" operator to implement the set membership. Returns true if an element is in the set.
- Overload the "+" operator to add an element to the set. Return the original set with the new element added.
- Overload the "-" operator to remove an element from the set. Return the original set with the new element removed.
- Overload the "+" operator to implement the union of two sets. Returns a new set that contains all the elements of the both sets.
- Overload the "*" operator to implement the intersection of two sets. Returns a new set that contains all the elements that are in both sets.
- Overload the "-" operator to implement the set difference. Returns a new set that contains all the elements that are in the first set but not in the second.
- Overload the "<=" operator to implement the subset. Returns true if all the elements of the first set are in the second.
- Overload the ">=" operator to implement the superset. Returns true if all the elements of the second set are in the first.
- Overload the "==" operator to implement the set equality. Returns true if both sets contain the same elements (in any order).
- A function called ***toString*** that returns a set string in the format `{1, 2, 3, 4}`). The set elements must be sorted.
- A function to return the number of elements in the set (***size***).
- A function to clear the set by removing all the elements (***clear***).
- Separate the MySet class into two files myset.h and myset.cc.

Write a main program to test your code or use the unit tests provided.

```
make run_tests
```

Hints: Follow these steps in order:

1. Design the class *MySet* with an array/vector of integers to store the numbers.
2. Write the *size* function.
3. Write the membership function (^).
4. Write the *+* functions.
5. Write the *toString* function.
6. **Test your class before proceeding with the rest of the functions. You may run the provided tests any time by issueing the command "`make run_tests`". It should test all the required functions.**
7. Write the clear functions.
8. Write the remove function (-).
9. Write the *intersection* function (*).
10. Write the *difference* function (-).
11. Write the *subset* function (<=).
12. Write the *superset* function (>=).
13. Write the *equal* function (==).

Grading:

Programs that contain syntax errors will earn zero points.
Programs that use global variables, other than constants, will earn zero points.

(60 points)

- 2 points - default constructor
- 5 points - for each of the functions: membership, adding an element, removing an element, toString, union, intersection, difference, subset, superset, and equality.
- 3 points - clear
- 5 points - Separate files

(10 points)

- Programming Style
- Documentation

Follow the coding style outline on GitHub:
https://github.com/nasseef/cs2400/blob/master/docs/coding-style.md