

CS 3358 (Data Structures, §003 & §005, Fall 2018) by Lee S. Koh

Assignment 1

(posted 8/27/2018, **DUE 9/12/2018**)

Preparation

Read [this supplement](#) for a quick refresher on set and some set operations.

Overview

Using the C++ `class` feature, design, implement and test a new (programmer-defined) data type that can be used to represent and manipulate sets of *integers*. The operations supported should include those mentioned in [this supplement](#).

Some Specifics

You will use a *fixed-sized, compile-time array* (and other supporting data members) to implement an `IntSet` data type (using C++ `class` with member variables declared `private`) that can be used to declare variables (objects) each of which can represent a set of up to `MAX_SIZE` *integers*. For the purpose of testing, set `MAX_SIZE` to 10; your design and implementation of the data type, however, should enable the maximum size to be easily modified, *i.e.*, only need to change the value that `MAX_SIZE` is set to.

Goal

To gain experience designing, implementing and testing a relatively simple programmer-defined data type using C++ `class`.

Relevant Textbook Material

Chapter 2.

Supplied Files

- [Starter files](#) - for uniformity, you MUST start with these.
- [Sample automatic test input/output files](#) - cases for testing invalid inputs or oversized sets *not* included.
 - Note that the output file was generated as follows (using *input and output redirection*):


```
./a1 auto < altest.in > altest.out
```
- [Sample executable](#)
 - Some caveats:
 - It was compiled using one of CS department's Linux machines. Thus, it may not run on machines that have different Unix operating systems (including different versions of Linux). It definitely will not run on Windows.
 - To use the file, you should first do the following: (1) download the posted file (`Assign01SampleExe.zip`) onto you local machine, (2) unzip the downloaded file to obtain `a1`, and (3) upload the unzipped file (`a1`) to the desired working directory on your Linux account.
 - You will most likely get an error message such as the one shown below when attempting to run the program (by entering `./a1` while in the directory containing the file):


```
bash: ./a1: Permission denied
```

If this happens, it is because the access permission for the file got changed during download/upload. To fix the problem, do the following while you are in the subdirectory containing the uploaded file (a1):

```
chmod +x a1
```

- Let me know as soon as possible (via e-mail) if you face any difficulties using the file, or if you discover some errors in my program.

Your Tasks

- Thoroughly study and make sure you *fully understand* the starter header file `IntSet.h` and the supplied test driver file `Assign01.cpp` – you shouldn't have to make any changes to this file. (Be sure to check with me first if you feel strongly that you have to make additions or changes to these files. To simplify the grading task as much as possible, there should not be any additions/changes to the two files unless there's indispensable need for them.)
- Implement all the functions whose specifications appear in `IntSet.h`. Start with the supplied implementation file `IntSet.cpp`, where the *invariant* (see pp. 105-106 of textbook) for the class and stubs for the other functions have been provided.
- Compile a test program using `IntSet.h`, `IntSet.cpp` and `Assign01.cpp`, and run the program using appropriate test cases. A *Makefile* has been included for your convenience.

Deliverables

- Source file: `IntSet.cpp`. (softcopy and hardcopy)
- Test run results (auto version). (hardcopy only)
 - You must *at least* test your implementation using the cases included in this [test input/output files](#). Be sure to check out this [explanation page](#) on how to use the file.
 - NOTE: During grading, your program will typically be tested much more thoroughly than the minimum cases required above.

NOTES:

- Be sure to **remove all irrelevant comments and code segments** (including any *debugging output statements*) in your `IntSet.cpp` when you submit your work.
 - This will help make my grading task easier. In particular, any extra output (from debugging output statements) will make it difficult for me to check your test output for correctness.
 - Be sure to read/follow the instructions on what/how to submit posted [here](#).
 - Refer to [this penalty tally sheet used in the past](#) to help you minimize losing points.
 - If you discover anything that appears incorrect, please let me know (through email) as soon as possible.
 - Be sure to check the homepage often for any further information related to this assignment.
-

Use your browser's **Back** button to go back to previous page. ([click here to return to my homepage](#))
