CS 341
Programming Language Design and Implementation
Fall 2020

Homework #02

Complete By: noon Friday 9/4
Policy: Individual work only, late work *not* accepted
Assignment: Using a set
Submission: submit electronically on Gradescope

Simple Application using set library

This assignment is to get familiar with the set data structure from the Standard Template Library (STL). Much of the code has been provided for you, with a few functions left commented out for you to fill in with an implementation using the set class. We will be reusing this framework on future assignments while examining the implementation of a set class and the modern C++ features utilized in the creation of such a class.

Random sequence

This assignment is to fill in the implementation of a C++ program which generates random numbers, adding them to a set, until a particular number has been found.

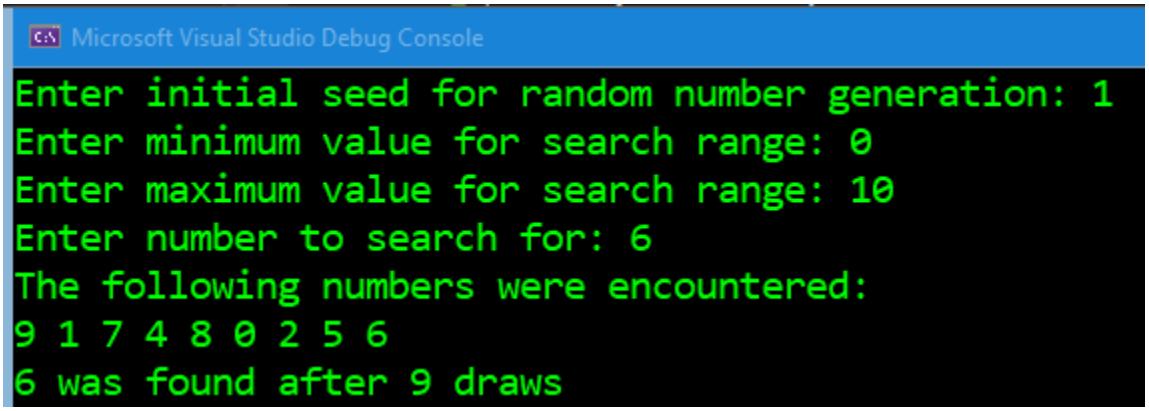
The program begins by taking in 4 inputs, a random seed, a minimum value of a range (inclusive), a maximum value of a range (exclusive), and a number to search for.

The program then proceeds with the search function, which you must implement. This function randomly generates numbers using the rand() function from the standard library, seeded with the first input, and adds them to a set until the target number is found. This last number is added to the set, and then the set is returned for further processing. The random numbers can be generated using the given randomInRange function, which includes the minimum value and excludes the maximum as stated in the description above. Using the set for this purpose means no duplicates are included in the set, and so the random generation behaves as drawing without replacement.

Then some further processing occurs. For this assignment, only two forms of analysis are included. The first is to output all of the unique numbers generated in the range. Since we are using unordered_set to store the numbers, we are guaranteed uniqueness of the elements. However, outputting them "in order" may not match the order they were generated, nor numeric order, and that is fine. The output will be compared against the default ordering produced by unordered_set.

For example, if the user were to input 1, 0, 10, 6, the random number generation would be seeded with 1, then numbers from 0-9 would be generated until a 6 is reached, with those numbers and the total number of generated values being output.

Here is a screenshot of the output in the first example given, though you may end up with a different sequence because the random number generation behaves differently on different systems, even with the same seed.



You should submit the source code containing your solution in a single file named main.cpp to Gradescope under Homework 2.

Piazza, not email

Do not use email for communication — all class-related emails will be ignored. To provide assistance, answer questions, and post announcements, we will be using a forum-based web site called *Piazza*. You should think of Piazza as your primary mechanism for all class-related discussions and questions. Piazza allows students and staff to help one another, reducing the *time-to-answer*. Your professor and TAs will check Piazza repeatedly each day — you should get in the habit of doing the same. Before posting, please search Piazza as the answer to your question may already be online. When posting, please follow these guidelines:

1. *Look before you post* — the main advantage of Piazza is that common questions are already answered, so search for an existing answer before you post a question. Posts are categorized to help you search, e.g. "Pre-class" or "HW".
2. Post publicly — only post privately when asked by the staff, or when it's absolutely necessary (e.g. the question is of a personal nature). Private posts defeat the purpose of piazza, which is answering questions to the benefit of everyone.
3. Don't post your entire answer — if you do, you just gave away the answer to the ENTIRE CLASS. Sometimes you will need to post code when asking a question --- in that case post only the fragment that denotes your question, and omit whatever details you can. If you must post the entire code (e.g. when asked to do so by one of the staff), then do so privately --- there's an option to create a private post ("visible to staff only").
4. Ask pointed questions — do not post a big chunk of code and then ask "help, please fix this". Staff and other students are willing to help, but we aren't going to type in that chunk of code to find the error. You need to narrow down the problem, and ask a pointed question, e.g. "on the 3rd line I get this error, I don't understand what that means…".
5. Post a screenshot — sometimes a picture captures the essence of your question better than text. Piazza allows the posting of images, so don't hesitate to take a screenshot and post; see http://www.take-a-screenshot.org/ for quick ways to take a screenshot on your platform of choice.

Other Policies

Late work is not accepted for this assignment. This assignment is mostly simple, but note that for assignments going forward all work is to be done individually — group work is not allowed. While we encourage you to talk to your peers and learn from them, this interaction must be superficial with regards to

all work submitted for grading. This means you *cannot* work in teams, you cannot work side-by-side, you cannot submit someone else's work (partial or complete) as your own. The University's policy is available here:

https://dos.uic.edu/conductforstudents.shtml .

In particular, note that you are guilty of academic dishonesty if you extend or receive any kind of unauthorized assistance. Absolutely no transfer of program code between students is permitted (paper or electronic), and you may not solicit code from family, friends, or online forums. Other examples of academic dishonesty include emailing your program to another student, copying-pasting code from the internet, working in a group on a homework assignment, and allowing a tutor, TA, or another individual to write an answer for you. It is also considered academic dishonesty if you click someone else's iClicker with the intent of answering for that student, whether for a quiz, exam, or class participation. Academic dishonesty is unacceptable, and penalties range from failure to expulsion from the university; cases are handled via the official student conduct process described at https://dos.uic.edu/conductforstudents.shtml .