

Project 1: Ordered List

Due: Tue, Feb 7

In this assignment, you will implement an `ordered_list` class. This class will allow the insertion and removal of integers into an ordered list (ascending). You MUST implement this using a linked list.

Getting started

Clone the project stub into the `Projects` subdirectory of SVN working directory.

```
cd ~/ID/Projects
svn export https://dev.cs.uakron.edu/svn/cs316sp17/shared/Projects/Ordered/
```

Be sure to `svn add` the newly exported `Ordered` directory and then commit.

Investigate the contents of that directory. You should have the following files:

- `CMakeLists.txt` — The build system for your homework. Read this file carefully.
- `ordered.cpp` — A small program that will utilize your `Ordered_list` class. DO NOT MODIFY THIS FILE!
- `ordered_list.cpp` — This is where you provide the implementation of `Ordered_list`.
- `ordered_list.hpp` — This is where you provide the definition of `Ordered_list`.

You can try building the project, but it won't work. The `ordered.cpp` file is written in terms of functions that *you* must provide.

Class requirements

The `Ordered_list` class must have the following:

1. A default constructor.
2. A destructor — memory leaks will cost you points
3. An `insert` function that takes an integer and inserts it into the list. There will be no duplicates in the list, if an integer is already present do not add another. This function returns true if the integer is in the list after `insert` is called.
4. A `remove` function that takes an integer and removes it from the list. It returns true if the integer was removed.
5. A `search` function that takes an integer and returns true if the integer is in the list.
6. An `is_empty` function that takes no parameters and returns true if the list is empty.
7. A `clear` function that takes no parameters and removes all elements from the list and returns nothing.
8. A `count` function that takes no parameters and returns the number of elements in the list.
9. A `display` function that takes no parameters and displays the contents of the list to the screen.
Each element should be surrounded by square brackets `[]`.

Define the `Ordered_list` class and declare all associated functions in `ordered_list.hpp`. Define all of those functions in `ordered_list.cpp`.

Testing

Create a build directory in the root directory of the project. DO NOT ADD THE BUILD DIRECTORY TO SVN! From this directory you should run the following code:

```
cmake ..
```

From this point you can type `make` inside the build directory to compile your code and `./ordered` to test the output of your code (once it compiles). The output of your program should match the `output.txt` file in the program directory. An easy way to check this is to run the following commands from the build directory.

```
make
./ordered > ../my_output.txt
diff ../output.txt ../my_output.txt
```

Submission

Homework is submitted in two ways:

1. Committing it to your SVN repository.
2. Submitting a printout on the day the project is due.

To generate the printout, simply type `make printout` in your build directory. This will generate a code listing in `printout.pdf` with the `ordered_list.hpp` and `ordered_list.cpp` files. Open the PDF (using Chrome, Firefox, or a PDF viewer) and print it (two-sided if possible). This is easily done from a lab computer. **Staple multiple sheets together.**

DO NOT FORGET TO BRING YOUR PRINTOUT TO CLASS.

Grading basis

If your homework is not in subversion OR you did not submit a printout, you will get a 0 on your assignment. You must submit **both** to receive a grade.

The total is out of 100 points.

- 75 You submitted code that compiles, and gives the correct output for the given input file yet does not give the correct output for additional test files I will be using.
- 90 You submitted code that compiles, and gives the correct output for the given input file as well as additional input files.
- You get 10 additional points for having no memory leaks. You can use `valgrind` to check for memory leaks on the knuth2 server (`valgrind ./ordered`).