

frankSJSU DataStructure

 Search this site

Frank's Home Page

CMPE126 home

Greensheet

Frank's Notes

[operator overloading](#)
[storing objects](#)
[pointer & deep copy](#)
[array of objects](#)
[linked list](#)
[variable size objects](#)
[create a linked node](#)
[create a linked list](#)
[linked list insertion](#)
[find middle](#)
[hybrid list](#)
[linked list quiz](#)
[recursion](#)
[stack](#)
[stack with array](#)
[math expression](#)
[queue](#)
[simulation](#)
[frankSimulation s16](#)
[priority queue & heap](#)
[search by hashing](#)

Frank's Slides

Frank's Code

Programming Exam

[PE #3 F16](#)
[PE #1 guide F15](#)

Midterm Exams

[midterm 2 F19](#)
[midterm 1 S18](#)
[midterm 2 F17](#)
[midterm 1 F17](#)
[midterm 2 S17](#)
[midterm1 S17](#)
[midterm2 F16](#)
[midterm1 F16](#)
[Labs and Homeworks >](#)

Lab 2 object array

Objectives:

1. Get yourself familiar with the file processing. Why? Because in real life, data are stored in files.
2. Use high level file IO as much as possible. Well, that's the essence of overloading << and >>
3. Importing is usually the hardest part of a database-like application because it needs to "convert" outside format to your internal design format.
4. If you're up to it, try to design two layers of classes: one for the object (complex) and the other for the collection of objects (complexDB)

Lab 2: File Import/Export and Object Array

Overview

Redo the complex homework and perform add / delete / list / save functions.

Read in a file of complex numbers (sample is attached **126import.txt**) and store them in an array of complex objects. (Would you be able to do either static or dynamic array?)

Create a menu to allow console add / delete / list and save functions. (You can also do load functions to make it more complete). The list and save function needs to print out the numbers in ascending order based on its value. The value of a complex number $a+bi$ is $\sqrt{a^2+b^2}$. Save function stores the results to file **126complex.txt**.

Note

If you can not handle a few corner cases, e.g. the last 3 complex number, remove them and focus on the main part. You will not get the full grade, but at least you need to get some code working.

You need to implement operator overloading to compare complex numbers as well as stream insertion / extraction operators.

It is highly recommended that you implement TWO classes (complex and complexDB) for this lab. It will be a good foundation for future data structure practices.

Bonus Points:

Use the challenge input file **126importHard.txt** to test for additional corner cases. Write your output to **126exportHard.txt** in the same program.

Lab 2b sorted object array

Objectives:

1. Repeat Lab 2 with two layers of classes: one for the object (complex) and the other for the collection of objects (complexDB)

- midterm S16
- Final Exams**
- Final S17
- Final S16
- Final F15
- Final S15

Labs and Homeworks

- Misc Lab FYI
- Lab 0 C++
- Lab 1 classes
- Lab 2 object array**
- Lab 3 Linked List
- Lab 4 Doubly Linked List
- Lab 5 Recursion
- Lab 6 Stack
- Lab 6+ math expression
- Lab 7 Simulation
- Lab 7a Palindrome
- Lab 8 search
- Lab 9 hashing
- Lab 10 sort

- 2. Overload less than **operator<** in complex class based on its value. The value of a complex number a+bi is sqrt(a*a+b*b).
- 3. Complex number is sorted in complexDB in ascending order such that complexDB can easily print in ascending order.



126import.txt (0k)

Frank sjsu Lin, Jan 28, 2...

v.2



126importHard.txt (0k)

Frank sjsu Lin, Jan 28, 2...

v.2



PRACTICE_INPUT.txt (0k)

Frank sjsu Lin, Jan 28, 2...

v.2



complex.txt (0k)

Frank sjsu Lin, Jan 28, 2...

v.2



Comments

You do not have permission to add comments.