

Lab 7

Strings, Vectors and Operators in C++

Objectives

1. Get more familiar with strings in C++
2. Learn about Vectors and Operators in C++

BRING YOUR TEXTBOOK TO THE LAB SESSION (Lippman). You will need it as a reference, and for some of the tasks that you will need to complete.

Part 1

In the first part of this lab you will work on the following example programs (posted on Blackboard):

1. Lab7_strings.cpp
2. Lab7_vectors.cpp
3. Lab7_operators.cpp

Study, compile and run the programs (start with _strings.cpp and finish with _operators.cpp). Each of the .cpp files has comments and instructions that you need to follow. There are several exercises that you need to carry out and report.

Part 2

In this part you will modify some of your classes and methods from Lab 6. The idea is to use strings, vectors and your own defined operators to achieve the corresponding tasks.

- Convert your signal data arrays into vectors. You may not need to dynamically allocate memory (and free it) any more.
- Modify your default empty constructor. Instead of using a default input data file, leave empty signal data vectors. Set the length and other data members to zero.
- Use strings wherever you can.
- Define (member) operators for offsetting (+) and scaling (*) your signals. Use those operators in your Centering and Normalization methods.
- Define a (non-member) addition operator that takes two Signal objects and returns a new Signal object with the following properties:
 - Only objects of the same length can be “added”.
 - The resulting signal data vectors should be the result of an element-wise addition.
 - The resulting maximum value data member should be the maximum among the two individual maximums (not the addition of the two).
 - What should the resulting average data member be?
- Depending on your lab 6 implementation, you may or may not need to modify your other methods a bit.

Push your code to your github repository.

Experiments:

For Part 1, run all the cases stated in the example files.

For Part 2, run similar experiments to those that you ran in Lab 6. Make sure you get the same results as before. Try your new addition operator several times. Make sure that the resulting object is as expected.

Deliverables:

1. Lab Report:

- 1) The report should follow the lab report format posted on Blackboard.
- 2) Results: show results of part 1 and 2. Include answers to questions that may have been presented in the example files (part 1). Include screenshots when appropriate.
- 3) Discussion: Make sure to discuss your results and to describe any problems and interesting things that you may have encountered while coding the lab, especially those related to the new concepts.
- 4) Screenshot of your Github repository showing your commit.
- 5) Append your source code with proper comments and indentation to the end of your report.

2. Demonstration:

Show a few test cases to the TA.

Grading:

<u>Demo:</u>	20	
<u>Report:</u>	40	
<u>Code:</u>	40	(Well organized, properly commented and indented)