

OPTIONAL PROGRAMMING ASSIGNMENT 4

OPERATOR OVERLOADING

DUE: Friday, April 28, 2017, Midnight
No Late Submissions Accepted

PROGRAM OVERVIEW:

This assignment is designed to give you practice with overloading operators. This is an **optional** programming assignment.

GRADING:

If you complete this assignment in accordance with the specifications, you will receive 10 points on one of your exams. This assignment is an **all or nothing** assignment. There will be no partial points and no extra credit. If you do not follow all instructions, you will receive 0 points. If your program does not compile you will receive 0 points.

LEARNING OBJECTIVES:

Completing this assignment will give you practice developing algorithms for overloaded operators and creating a makefile

ACADEMIC INTEGRITY:

This is an individual assignment. You may not receive help from anyone other a lab TA. I will not provide help with this assignment. Please review the academic integrity policy provided in the syllabus. All academic integrity policies apply.

REQUIREMENTS:

You are required to implement the class defined in **MyClass.h**. I will provide the **MyClass.cpp** file that will contain two client functions already implemented for your use. You are required to implement the remaining functions listed in the header file. These functions should be implemented in the **MyClass.cpp** file. I will also provide a driver for you to use to test your functions, as well as an **input.txt** file. I will provide an **output.txt** file that you will use to compare your output. Your output **must** match that of the provided output.txt file. When testing, I will test your program using two drivers: the driver provided and one of my choice. You are required to provide a **makefile** (which should produce an executable named **driver**), as well as a **readme** file. Your readme file should describe any problems you had with the assignment, how you solved the problems, and your thoughts on the assignment.

OPTIONAL PROGRAMMING ASSIGNMENT 4

OPERATOR OVERLOADING

DUE: Friday, April 28, 2017, Midnight

No Late Submissions Accepted

This program will use redirection rather than command line arguments. To run your program you should type the following:

./driver < input.txt >out.txt

The above will produce an out.txt file with the output of your program. You should compare your output with that of the output.txt file I provided.

FORMATTING:

You will need to add a header to each of your files similar to the following:

```
/******  
*Your name  
*CPSC 1020 your section, Sp17  
*Your user name  
*****/
```

- Your code should be well documented. (comments)
- There should be no lines of code longer than 80 characters.
- You should use proper and consistent indentation.

HANDIN:

Use handin.cs.clemson.edu to submit your files. I have created buckets named PA4.

Things to do prior to handing in your files:

1. **Test your program on the SoC servers.** I will not accept the excuse "It compiled on my computer." I test programming assignments on the SoC servers.
2. Tar zip your files naming the tarred file PA4.tar.gz When I run untar your files should be there. I should not have to change directories multiple times to get to your files. This will break my grading script.
3. The files you need to hand in should be the following:
 - a. driver.cpp
 - b. makefile
 - c. MyArray.h
 - d. MyArray.cpp
 - e. input.txt
 - f. readme