

Programming Project Report

Name: Logan Campbell

Date: 7/5/2017

Academic Integrity Statement: I pledge that I have neither given nor received unauthorized help on this programming assignment.

Problem Statement:

The main goal for this programming assignment is to use object oriented programming and the use of classes to define a header file to use in a main program. Defining around 23 functions, these functions are used to declare, adjust, clear, and modify a matrix. Dynamically that is... Error checking would have to be the bounds of the matrix and input would have to be the function inputs.

Design:

The design of the program was already made which was a main program involving a header file. The header file that contained object oriented implementations. Having the template of the header file is all that was needed to complete the function. In which making the main program run smoothly with each function in the matrix class. The only con that I see with this is that if something is wrong with your header file, everything becomes messed up within your header file and your main file.

Implementation:

The sample code is the skeleton of the header file and the complete usage of each function in the main program. The sample code is quite a bit, but to sum it up it's a matrix class. Extending the code required the implementation of 23 different functions. Starting with each numbered function in the homework guide. The program can be tested by doing each function at a time and running the main program. Doing this ensure a complete functional program, that wouldn't work because of a procedure missed or an incorrectly implemented procedure.

Testing:

I tested the program by seeing what would happen if I input values that were either out of bounds or different data types. Usually it would return the special cases of such inputs, however, for the different data types: strings and bools wouldn't compile. For the most part, it would convert any other data type to an integer value since the program is dealing with a set of elements for a matrix. In return, making all the input information be converted to usable type for the header and main program. I have around 4 pictures attempting different data types and inputs...

- Attempt for negative inputs
- Attempt of a character data type
- Bounds checking of get cell/ problem fix
- Bounds checking of other functions like mean

Conclusions:

Overall the project required the usage of classes and a header file to incorporate object oriented programming to a main program. The project was about creating an array of elements that act as a matrix and additionally the different functions to adjust and modify it. For the most part, the program was smooth. I had trouble dealing with the copy constructor, dealing with certain parameters of a 1-dimensional array that is converted to a two-dimensional array, and the resize

function. Each was solved by either setting up the correct loop, or the copying of the correct information. I would beware of the bounds before creating the program rather than developing the program and then deal with bounds. Because of that, my code is misleading on the actual functionality when it shouldn't be dealing with certain conditions in the first place. Furthermore, project took about 3 or 4 days.

```
// Window Statistics
```

```
cout << "Testing Window Statistics" << endl;
```

```
cout << "Sum:      " << m2.sum(-2, 2, 5, 6) << endl;
```

```
cout << "Product:  " << m2.product(-2, 2, 5, 6) << endl;
```

```
cout << "Max:      " << m2.max(-2, 2, 5, 6) << endl;
```

```
cout << "Mean:     " << m2.mean(-1, 2, 5, 6) << endl;
```

```
cout << endl;
```

```
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
```

```
Testing Window Statistics
Sum:      0
Product:  0
Max:      0
Mean:     0
```

```
// fill()
cout << "Testing fill()" << endl;
m2.fill('A');
m2.print();
cout << endl;
```

```
0 0 0 0 0
0 0 0 0 0
Testing fill()
65 65 65 65 65
65 65 65 65 65
65 65 65 65 65
65 65 65 65 65
Testing setCell()
```

1	-1	-2	-3	-4
1	3	-1	-2	-3
2	1	5	-1	-2
3	2	1	7	-1

Testing getCell()

-1

0

268496024

0

0

Testing Matrix Statistics

Sum: 6

Product: 362880

Max: 7

Mean: 0.3

Testing resize()

1	-1	-2	-3	-4	0	0
1	3	-1	-2	-3	0	0
2	1	5	-1	-2	0	0
3	2	1	7	-1	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

```

Testing fill()
  5    5    5    5    5
  5    5    5    5    5
  5    5    5    5    5
  5    5    5    5    5

Testing setCell()
  1   -1   -2   -3   -4
  1    3   -1   -2   -3
  2    1    5   -1   -2
  3    2    1    7   -1

Testing getCell()
-1
0
0
0
0

Testing Matrix Statistics
Sum:    6
Product: 362880
Max:    7
Mean:   0.3

Testing resize()
  1   -1   -2   -3   -4    0    0
  1    3   -1   -2   -3    0    0
  2    1    5   -1   -2    0    0
  3    2    1    7   -1    0    0
  0    0    0    0    0    0    0
  0    0    0    0    0    0    0

Testing addRow() / addCol()
  1   -1   -2   -3   -4    0    0    0
  1    3   -1   -2   -3    0    0    0
  2    1    5   -1   -2    0    0    0
  3    2    1    7   -1    0    0    0
  0    0    0    0    0    0    0    0
  0    0    0    0    0    0    0    0
  0    0    0    0    0    0    0    0

Testing Window Statistics
Sum:    9
Product: 0
Max:    7
Mean:   0

Testing Copy Constructor
- Original:
  1   -1   -2   -3   -4    0    0    0
  1    3   -1   -2   -3    0    0    0
  2    1    5   -1   -2    0    0    0
  3    2    1    7   -1    0    0    0
  0    0    0    0    0    0    0    0
  0    0    0    0    0    0    0    0
  0    0    0    0    0    0    0    0

```