# CSCI 2312-001: Object Oriented Programming

# Assignment 2

# Assigned:

# Due: at 11:59 PM

**Instructor: Marlu Oswald**

**Email: marlu.oswald@ucdenver.edu**

**Assignment 2**

Remember (ROCkET programming methodology!)

Build two grids which are each X wide and Y long where X and Y are user inputs to determine the size of the grids. Fill each cell of both grids with a ‘0’. Randomly fill 1/3 of the grid’s cells with a ‘1’. You must randomly fill each of the two grids separately so they do not have the all of the same squared filled with a ‘1’. Compare the two grids (square by square comparison) to find squares which have a ‘1’ in both grids. Create a third grid which contains a ‘1’ in squares where both of the compared grids contain a ‘1’ and has a ‘0’ in any other square.

Example in a 2 by 2 grid:

Column1 (Y) Column2

Row1 ( X)

Row2

|  |  |
| --- | --- |
| 0 | 0 |
| 1 | 1 |

|  |  |
| --- | --- |
| 1 | 0 |
| 1 | 0 |

Grid 1 Grid 2

|  |  |
| --- | --- |
| 0 | 0 |
| 1 | 0 |

Grid 3 (Result)

For this assignment, you will need to create two grids using the Vector STL (Standard Template Library) and the appropriate functions to perform the above operations. Some basic level of error checking is also required for full credit.

Additionally, you will need to submit readable debugger screen shot(s) zoomed in on the debugging window to show the values of both grids when the user enters a 2 by 2 grid size.

**Random Numbers**

Example program. Remember that you only seed the random function once during an execution!

#include **<iostream>**#include **<cstdlib>** *// srand, rand*#include **<ctime>** *// time***using namespace** std;  
  
**int** main ()  
{  
 **int** randNum;  
 */\* initialize random seed only once in an execution \*/* srand (time(**nullptr**));  
  
 **for** (**int** i = 0;i < 5; i++)  
 {  
 */\* generate a number between 1 and 10: \*/* randNum = ( (**int**)rand() % 10) + 1;  
 cout << randNum << endl;  
 }  
 **return** 0;  
}

**Grading Rubric**

Follow best practices in programming with declarations in .h files and implementations in .cpp files. Ensure your name is on every file in comments. Update your makefile (not makefile.txt!) AFTER you have tested successfully on the csegrid, update your readme.txt file then move over to your system. Create a folder named lastnameOB1 and place (just) your .h, .cpp, readme.txt, and makefile in that folder. Zip the folder and turn it into Canvas by the due date.

You may (and are encouraged to) discuss general strategy with classmates (or other help) but you MUST document all extra help other than the textbook (including web sites) with comments. Failure to document extra help may result in a zero for the assignment. While you should help your classmates, stop short of giving exact lines of code or pseudocode. If we can’t determine who shared and who copied, we may give you both a zero for the assignment.

|  |  |
| --- | --- |
| Correct and Complete Functionality | 12 |
| Compile and Fault Tolerance (on csegrid) | 2 |
| makefile and readme.txt | 2 |
| Documentation / Comments | 2 |
| Debugging Screen Shot | 2 |