**CS1220 – C++ Programming**

Homework Assignment #5

Due: See [course web site](http://people.cedarville.edu/Employee/kshomper/cs1220_web/schedule.htm) for due date

# Points: 80

**Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

I. Requirements: Restate the problem specification, and any detailed requirements

II. Design: How did you attack the problem? What choices did you make in your design, and why? Show class diagrams for more complex designs.

III. Security Analysis: State the potential security vulnerabilities of your design. How could these vulnerabilities be exploited by an adversary? What would be the impact if the vulnerability was exploited?

IV. Implementation: Outline any interesting implementation details in your solution.

V. Testing: Explain how you tested your program, enumerating the tests if possible. Explain why your test set was sufficient to believe that the software is working properly, i.e., what were the range of errors for which you were testing.

VI. Summary/Conclusion: Present your results. Did it work properly? Are there any limitations? NOTE: If it is an analysis-type project, this section may be significantly longer than for a simple implementation-type project.

VII. Lessons Learned: List any lessons learned. For example, what might you have done differently if you were going to solve this problem again?

**Objective**

Build a tic-tac-toe game using wxWidgets and (optionally, but strongly recommended) wxFormBuilder.

**Description**

The purpose of this assignment is to have some fun with graphical user interface (GUI) development while learning more about C++ inheritance and event programming. To get started, you may want to play the tic-tac-toe game (TTT.exe) located at *S:\DEPT\EG\Computer Science\CS1220\Tic Tac Toe.*

Note: if you want to use wxWidgets and wxFormBuilder on your laptop, follow the setup instructions ([here](http://people.cedarville.edu/Employee/kshomper/cs1220_web/developing_wxwidgets_with_visual_studio_and_wxformbuilder.docx)).

The tic-tac-toe game should be played as demonstrated by the example program. More specifically, you game should provide

(1) A 3x3 grid of game-play buttons,

(2) Reset and exit buttons having event handlers to take these actions,

(3) An “exit” menu option also with an event handler,

(4) A status bar which lets the players know who has the current turn,

(5) Appropriate dialogs to confirm or cancel all reset and exit actions,

(6) Recognition and announcement when a player wins or ties, and

(7) After a game concludes, a dialog should appear requesting replay.

**Other Details**

1. You are to develop the program individually as we have with all other assignments (In the past we permitted two students to work together on this project, but that policy is NO longer in effect).
2. Grading for this assignment will be a bit different than in the others, in that you will receive a third of the points (30 total) simply for

1. Building a full user interface (i.e., all required buttons and menus) and

2. Getting the “Exit” and “Reset” menu and button options to work.

The remainder of the points will be allocated to getting the game play and the rest of the wxObjects working correctly.

1. Turn in a printed listing of *only* the source file SubGUIClass.h/cpp.

Required for turn-in:

1. The completed coversheet for this assignment.
2. A printed listing of your event handlers (i.e., the files SubGUIClass.h and SubGUIClass.cpp). Your code should be properly commented, and in compliance with Cedarville C++ style guide.
3. NOTE: your solution should compile, link and run on john, so please verify this condition before submitting your solution. The last page of the [wxWidgets development notes](http://people.cedarville.edu/Employee/kshomper/cs1220_web/developing_wxwidgets_with_visual_studio_and_wxformbuilder.docx) provides instructions for (if necessary) modifying your solution to run on john.
4. An electronic zipped submission of ALL your source files (including main program)via the “submit20” command (demonstrated in class).