Assignment #1 – Review Programming in C++
Due: Sunday, 04/12/15, 11:59pm

Grading: For each programming assignment, you are graded by explaining and demoing your code to a TA. **You must demo your program BEFORE the next assignment is due**, and if you fail to do so, you will automatically lose 50 points! **Your job is to convince the TA that your program works correctly, i.e. show your TA how to use/break your program☺**

(80 pts) **Problem Statement:** Write a C++ program that plays the game of connect four. To see how to play the traditional 2-person game, visit page 2 of these instructions: http://www.hasbro.com/common/documents/dad2614d1c4311ddbd0b0800200c9a66/1EF6874419B9F36910222EB9858E8CB8.pdf

Of course, our connect four is going to have a slight twist! We will support the traditional, 6 x 7 board with connecting four horizontally, diagonally, and vertically, but we will also support an n x m board with connecting p pieces. You will take these values as command-line arguments with these options –r, –c, and –p to indicate rows, columns, and pieces, and these option/value pairs can be provided in any order. You need to make sure you handle inappropriate data, i.e. non-positive integers, and ask the user again for the correct input. You do not have to recover from bad options. You must continue to play the game, until the user no longer wants to play, without creating a memory leak!!!!

Example:

```
./connect_four –r 6 –c 7 –p 0

You cannot have 0 pieces to connect.
Please enter a positive, non-zero integer for the number of pieces to connect: 4
Player one, do you want red or yellow (r or y)? r

|  |  |  |  |  |  |  |
------------------------
|  |  |  |  |  |  |  |
------------------------
|  |  |  |  |  |  |  |
------------------------
|  |  |  |  |  |  |  |
------------------------
|  |  |  |  |  |  |  |
------------------------
|  |  |  |  |  |  |  |
------------------------
 Player 1, what column do you want to put your piece? 4
```

```
|  |  |  |  |  |  |  |
------------------------
|  |  |  |  |  |  |  |
------------------------
|  |  |  |  |  |  |  |
------------------------
|  |  |  |  |  |  |  |
------------------------
|  |  |  |  |  |  |  |
------------------------
|  |  |  | r |  |  |  |
------------------------
 Player 2, what column do you want to put your piece? 5

|  |  |  |  |  |  |  |
------------------------
|  |  |  |  |  |  |  |
------------------------
|  |  |  |  |  |  |  |
------------------------
|  |  |  |  |  |  |  |
------------------------
|  |  |  |  |  |  |  |
------------------------
|  |  |  | r | y |  |  |
------------------------
 Player 1, what column do you want to put your piece? 4

|  |  |  |  |  |  |  |
------------------------
|  |  |  |  |  |  |  |
------------------------
|  |  |  |  |  |  |  |
------------------------
|  |  |  |  |  |  |  |
------------------------
|  |  |  | r |  |  |  |
------------------------
|  |  |  | r | y |  |  |
------------------------
 Player 2, what column do you want to put your piece?...

…

Do you want to play again (0-no, 1-yes)? 0
```

For this assignment, you need to have a game struct that contains a pointer to the board put on the heap and the players' piece selection, i.e. 'r' for red and 'y' for yellow.

```
struct game {
  char **board;
  int r, c, p;
  char p1;
  char p2;
};
```

Another guideline is that you must have functions that are 10 lines or less. At a minimum, you must have these functions:

```
bool is_valid_arguments(char *info[]);
void set_game_info(game *, char *info[]);
char** create_table(int, int);
void play_game(game *);
bool check_winner(game);
void delete_table(game *);
```

You need to separate your files into interface and implementation and create a **Makefile** to handle the compilation. Create a **connect_four.h**, which has the struct type and the function declarations for your program. Now, separate your function definitions into a **connect_four.cpp** file and your main function into a **play_cf.cpp** file. Now, create a makefile that will create a connect_four executable game and clean your files.

Your program must be able to:

- Print a usage message to the user when too few arguments are supplied or when the options are not –r, -c, or –p. You do not need to recover from this, just handle by printing a message.
- Print an error message and recover, when the user doesn't supply positive, non-zero integer for the option values.
- Print an error message and recover, when the player doesn't supply a valid column on the board. This includes selecting a column that has filled all rows.
- Play the game correctly based on how many pieces to connect or when any user fails to win by the board completely filling.
- Continue to play until the user selects no. Make sure you do not have a memory leak!!!!!

(10 pts Extra Credit) **Computer against user connect four!!!** You can get 10 points extra credit if you write some type of Artificial Intelligence used for the computer to play one player in a game of connect four. You can make this as easy or as hard as you want☺

(10 pts) **Program Style/Comments**
In your implementation, make sure that you include a program header in your program, in addition to proper indentation/spacing and other comments! Below is an example

header to include. Make sure you review the style guidelines for this class, and begin trying to follow them, i.e. don't align everything on the left or put everything on one line!
http://classes.engr.oregonstate.edu/eecs/spring2015/cs162-001/162_style_guideline.pdf

```
/*********************************************************
** Program: play_cf.cpp
** Author: Your Name
** Date: 04/08/2015
** Description:
** Input:
** Output:
*********************************************************/
```

(10 pts) **Design for Assignment #1 changes/Testing**

- **(5 pts) How did you design for Assignment #1 change during implementation?**

- **(5 pts) What were the actual values from your testing? Did these match your expected values? What did you do to make sure you get the expected values?**

  Please see the template for this document: Polya_template.pdf
  You need to have a table with these headings:

Input Values	Expected Output	Did Actual Meet Expected?
-r 6 –c 7 –p 0	Error message for zero pieces and re-prompt for number of pieces to connect	Yes

Electronically submit your C++ program (**.h, .cpp, and Makefile files**, not your executable!!!) and design/testing document, **as a pdf**, by the assignment due date, using TEACH.

****NOTE:** The easiest way to upload your program from ENGR to TEACH is to map a network drive to your home directory on ENGR. Mac or Windows, See:
http://engineering.oregonstate.edu/computing/fileaccess/