Quiz Group 10 (C. Hash, K. Marino, K. Root, E. Rouse)
CS161 – Assignment #5/Quiz #10
August 1, 2012

Q1: Discuss Assignment #5, and answer the questions below:

- **write the pseudocode for the algorithm you are going to use to check for a winner in tic-tac-toe. How would you decompose the checkForWinner() method to ensure that it is under 20 lines of code;**

There are three ways to win tic tac toe. You can get three of the same letter in a row, you can get 3 in a column, or you can get three diagonally. Each of these can be checked with a for loop. Check all three in succession and BAM you got it.

checkForWinner()

```
FOR each row:

    is the first element equal to the other two elements and not an empty cell?
        if yes, return TRUE

FOR each column:

    is the first element equal to the other two elements and not an empty cell?
        if yes, return TRUE

 FOR each diagonal:

    is the center element equal to the other two elements and not an empty cell?
        if yes, return TRUE

return FALSE by default, which will only happen if none of the other checks return true.
```

- **Describe how you plan to play the hangman game. What will you do when the user enters a letter they have already chosen, i.e. will you print a message reminding the user that they have already chosen that letter and/or will this count against the number of guesses? Will this decision differ if the guess was found versus not found? In addition, describe the states and methods you need in the game of Hangman;**

I chose to ignore multiple entries of the same letter so as to not overly penalize a player. It doesn't differ when found or when not found. It seems a bit harsh to me and I don't like it when people play that way, so I let it go.

Methods required:

1. Default Constructor
2. Overloaded Constructor
3. determinePlayerChoice() to let the player choose their name
4. setWord() to get Player 1 to enter their word for player 2 to guess
5. guessLetter() lets player 2 guess a letter
6. compareGuessToWord() this takes care of the task of finding the guess in the word and replacing the dashed representation with letters to show player 2. If the guess is not in the word it is added to a list of all the bad guesses.
7. checkForWinner() checks to see if the words are the same. If so player 2 won.
8. printWinnerResults() print the results of a player 2 win
9. checkForLoser() checks to see if all the guess are used. If so, player 2 lost, player 1 made an unguessable word.
10. printLoserResults() prints the player 2 lost message.
11. printError() prints invalid response message to faulty input.