

CS11 Programming Assignment # 5 due Thursday 28 Oct. 1999

One-Word Hangman Game

Write a program that lets the user play a simplified version of hangman. In this game, the user guesses the letters of some mysterious word one at a time. The game ends when the player has correctly identified all of the letters in the word or has made too many incorrect guesses for letters. The purpose of this assignment is to become familiar with using strings in C++.

You should be familiar with the semantics of the following built-in functions: *isupper*, *islower*, *toupper*, *tolower* from `ctype.h`, and *strlen*, *strcmp*, *strchr* from `string.h`. You may find these functions useful in developing the program.

Your program should be written so that in order to change the solution word it is only necessary to retype a single string in the code. Similarly, changing the threshold number of bad guesses should be accomplished by changing only one number in the source code. I will run your program with different solution words and thresholds. Therefore, the length of the solution word needs to be calculated in your program, rather than hard-coded as a constant. You may assume that the solution word will never contain more than 30 letters.

Begin the game by showing the user a string of underscores equal in length to the mystery word. Ask the user for a letter. Notice that the solution word consists of lowercase letters. You may assume that the user will not enter capital letters. If the letter the player guesses appears in the word, reveal those locations by replacing the underscores with the correct letter. Note that it is possible for a letter to appear several times in the word. If the user guesses a letter that does not appear in the word, increment the number of bad guesses by one. If the player enters a letter that has already been guessed, it is automatically a bad guess.

See the example I/O on the next page. Your output does not have to look exactly like mine, but the general idea is that on each turn the user should be able to tell at a glance how close he is from winning or losing the game.

Example I/O:

```
*****
*           Welcome to the Hangman Game           *
*****

Your solution thus far: _____                0 bad guesses
Enter your guess: i

Your solution thus far: _____i_i__          0 bad guesses
Enter your guess: s

Your solution thus far: s_____i_i__          0 bad guesses
Enter your guess: n

Your solution thus far: s___n_i_i__             0 bad guesses
Enter your guess: r

Your solution thus far: s_r_n_i_i__            0 bad guesses
Enter your guess: l
Sorry, that letter is not in the word.

Your solution thus far: s_r_n_i_i__            1 bad guess
Enter your guess: t

Your solution thus far: s_r_n_i_it_            1 bad guess
Enter your guess: e

Your solution thus far: seren_i_it_            1 bad guess
Enter your guess: y

Your solution thus far: seren_i_ity            1 bad guess
Enter your guess: s
You guessed that letter already!

Your solution thus far: seren_i_ity            2 bad guesses
Enter your guess: d

Your solution thus far: serendi_ity            2 bad guesses
Enter your guess: p

Your solution thus far: serendipity            2 bad guesses

*****
**** You win ****
*****
```