Aryan Aziz

Assignment 5 Report

Exercises

List of Files:
assignment5-1.cpp
assignment5-2.cpp
pp4-17.cpp
pp13-1.cpp
pp13-8.cpp
pp13-10.cpp

Projects

Understanding –
So, for this project, the basis behind it seems to be a hangman game. And, although very simple in real-world ideas and playing, becomes a lot more complex than I thought in the programming world.

The game is broken up into 2 players and is defined as 1 player putting in a word and the other subsequently guessing x amount of letters to get the word. The program should keep track of what letters the user guessed, as well as how many guesses remain and the word/guessed letters in the word itself.

The program should check to make sure the word is valid and if the user has guessed the letter before or not.

Design –
Attached to this pdf are my designs for all programing projects and main project.

I decided to design the game a little differently. Instead of displaying the characters the user is able to guess, I decided to display the characters the user has already guessed (very similar to normal hangman).

I also decided to make the total amount of guesses equal to 2x the length of the word. Because of this, I felt like the words shouldn't be too long because the guesses would exponentially grow and decided to restrict that to 10 letters.

I also decided to re-use my do-while loop that I have used on previous projects for the play again. I'm still not sure if this is the best or most elegant solution, but it works and that's all I can really ask at this point.

Testing –
As usual, with my lack of confidence in my code, I pretty much compiled and tested each line as I went along. I tested to make sure that the word doesn't allow any special characters and as long as there is anything other than "abcdefghijklmnopqrstuvwxyz" in the string it will immediately prompt the user for another guess.

What took the longest amount of testing was actually the number of guesses and counting that correctly. I hit a snag that was causing the guesses to not count down correctly and was giving an extra guess to the user. After, literally, 3 hours of attempting and attempting I finally realized that I was putting the counter at the bottom of a loop and I was doing other things at the top of the loop before the counter (effectively putting the whole game 1 guess behind).

Reflection –
There are 2 huge things I learned from this week's assignment.

1. Turning real-world items in viable programs is a LOT more difficult than you would think.
2. Knowing what items to use/how to create the item from #1 is even more difficult.

In theory, the idea of creating a hangman game seems so easy. Take in a word, ask the user to guess, and see if they get it (similar to the number guessing game we made before). But unfortunately, dealing with strings is 10x more difficult than dealing with integers and brings in a whole other wealth of problems. (Mo' variable types, Mo' problems…….I got 99 problems, and they all strings) I got caught in a spot where I was attempting to check a char vs a string which was giving me huge problems.

What I thought would actually be a simple assignment ended up being the longest assignment to do because if became a lot more intricate and growing upon itself. I got some recursion in this baby (oh yea) but it took me quite a bit to get that integrated.

And as I was speaking about in the testing, I was literally pulling my hair out at the counter. I had the counter originally at line 76 (at the end of the while loop) and it was counting 1 behind (meaning they would get 10 guesses instead of 9, etc). It took me quite a bit to try to put the counter directly into the playerGuess function, which then FINALLY fixed my issue.

CS135 Assignment 5 4/19/14

- pp 13-8

input: initial $

interest rate

# of years

function

Compound interest = ($(interest) + $1) (a) — a(interest) + a (b) ...

$= P(1+i)^n$

P = $

i = interest

n = year

CSIBS Assignment 5 4/19/14

- pp 13.1

Recursive function

1 param (int)

returns ↑ fib #

fibonacci = 1, 1, 2, 3, 5, 8, 13...
(0 1 2 3 4 5 6)

fib(n) = fib(n-2) + fib(n-1)

Program starts
↓
user inputs n value
↓
fibonacci function
↓
If position 0 or 1
↓
return 1

- pp 13.10

recursive function

1 param (int) = # of rows

return total # of pins

prog start
↓
user inputs n value

Each row = row # of pins

N = N pins

If 1 = 1 2 = 2 3 = 3...

# of pins = n + (n-1) + (n-2) + ...

n = 1 then pins = 1

return 1

otherwise return (n + (n-1))

CS135 Assignment 5 4/18/14

° pp 4-17

- function: sort
  - input 3 int params by reference
  - switch variables around so that a gets lowest value, etc.

Program Starts
↓
User inputs 3 values
↓
Sort function
↓
find highest value
↓
find lowest value
↓
grab remaining value
↓
Set A to lowes
↓
set b to middle
↓
Set c to higres

CS 105 Assignment 5

4/19/14

- hangman
- player 1 enters a word
  - makes sure the word is valid
    - no symbols, spaces, or numbers
- displays # of letters + remaining guesses
- player 2 guesses a letter
  - sets it on "used letter list"
  - checks if letter exists in word
  - removes a guess

loop