Last updated 09 October, 2014

Assignment 3 (100 points):

Please see Submission Checklist (below) for submission requirements.

It is time to bring together some concepts we have been learning!

Keep in mind that there is more than one way to solve most things with programming.

(10 points) Programming Styles and Convention.

http://classes.engr.oregonstate.edu/~jessjo/CS161/OSU_IntroCodeStandards_v1.1.pdf

(30 points) Remember your project report!

Which includes:

- Understanding: Requires a description of what you understand is being asked of you for the assignment.
- Design: The design section of your report is required to be done before you do any coding. It is an outline in pseudo code **or a drawing** as to how your code will be implemented. This must include **both** the exercise and project component (if there is a project for the week).
- Testing: The testing section of your report is required to be done before you do any coding. You are to describe the tests you will implement to ensure your code works along with the actual testing. This must include **both** the exercise and project component (if there is a project for the week). **(Make a table with three columns named Input, Expected Output and Actual Output. Discuss any discrepancies.)**
- Reflection: For the reflections section you need to discuss the problems you encounter and how you solved them.

(20 points) Exercise components:

1. (4 points) Work with simple strings, create a program with the following:

File must be called: strings.cpp

a. Include the string library (**#include <string>**),

b. create a string variable (**string my_str;**),

c. get user input for that string (**cin >> my_str;**),

d. use a for loop to print the string one character at a time using the at function

(example: **for(int i = 0; i < my_str.length(); i++){cout << my_str.at(i) << endl;}**)

e. use a for loop to print the string one character at a time in reverse using the at function

(example: **for(int i...){cout << my_str.at(i) << endl;}**)

Discuss on Discussion Board: notice how the body in my example for loop looks the same but that I hide the for loop's definition, how could I use this to make this print the string in reverse?

f. use a loop to count how many letters there are in the word, keeping in mind that not all characters in the word need to be letters...

Note: this exercise step asks you to count the number of letters in the word rather than the number of characters that are in the word.

Discuss on Discussion Board: the actual solution to the above is up to you, but let others know how you did this!

g. Find out how you might be able to do this with multiple words in the input!

   (**hint**: using the getline function is one way)
   (example: **getline(cin, my_str);**)

   Discuss on Discussion Board: what happens if you extract from cin and use getline in the same code repeatedly? How can you fix this?

2. (4 points) Write a program.

   File must be called: name.cpp

   "Write a program that reads a person's name in the following format: first name, then middle name or initial, and then last name. The program then outputs the name in the following format: Last_Name, First_Name, Middle_Initial.

   For example, the input

   Mary Average User

   should produce the output

   User, Mary A.

   The input

   Mary A. User

   should also produce the output

   User, Mary A.

   Your program should place a period after the middle initial even if the input did not contain a period. Your program should allow for users who give no middle name or middle initial. In that case, the output, of course, contains no middle name or initial. For example, the input

   Mary User

   should produce the output

   User, Mary

   If you are using C-strings, assume that each name is at most 20 characters long. Alternatively, use the class string. (Hint: You may want to use three string variables rather than one large string variable for the input. You may find it easier to not use getline.)"

   (**hint**: I personally still find it easier to use getline...)

   Discuss on Discussion Board: how might we be able to solve this without using the getline function?

3. (4 points) Find out how to use a simple random number:

   File must be called: randNum.cpp

   a. Visit http://www.cplusplus.com/reference/cstdlib/rand/, my sample code directory, the tutorials I mention, and any searches you want (including the book if you can find something good in there :)),

   b. Print some random numbers using a loop (say 10 of them),

      Discuss on Discussion Board: Run the program a few times and see what happens, did you get the same numbers multiple times? What is happening here? Can we fix this?

4. (4 points) Write a piece of code using a loop that accepts two strings and returns whether they have the same contents with the use of a loop to check each character individually,

   File must be called: loop.cpp

   (**hint**: you can compare individual characters with the <, ==, and > operators)

   Discussion idea on Discussion Board: is there an easier way to check for equality of two strings? Why might I require you to "do it the hard way"?

5. (4 points) Write a piece of code that acts like a random number generator with the following behavior:

   File must be called: randNum2.cpp

   (**Note**: you can use the rand function in your function, you do not have to make your own random number generator code from scratch!)

   a. it should ask for two int values (one for the minimum value for some random number in a range and one for the maximum number in the range),

   b. check that the values entered were int values,

      (**Note**: there are a number of ways user input can be incorrect, so think of a few ways input could be incorrect and implement them in code if you can)

      Discuss on Discussion Board: What ways did you think of that user input could be incorrect for this problem, how might you be able to detect and possibly correct those errors, and were you able to implement it?

   c. generate a number within that range,

   d. assign the number generated to some variable,

   e. and print out the value along with some explanation so the user knows what the value is for (you will not always want to print values out, but in this case we want to see that our code is working)

      Discuss on Discussion Board: I am guessing you may find ways to do this with a one line expression, so feel free to share why it might be a good idea or bad idea to have this as a few separate lines instead of as a one line expression in our code?

6. Remember to discuss the report sections on the discussion boards, these skills will be **very** important later on!

(40 points) Project Component (design only):

You need to be able to spend some time designing projects before you start programming for them, so this assignment's project will be a design for a word guessing game.

1. (10) Start a design based on what we know about user input, variables, if statements, and loops for a word guessing game. What would your word guessing game behave like? You can think of this like a hangman game, a Wheel of Fortune game, or other similar word guessing game.

   Your response to this should contain some sort of **explanation or drawing** for what you would create for a word guessing game if it were completely up to you.

   (**Note**: if you have not learned anything about strings yet, then you can simply write directions such as "get word as input", "check that word is valid", "get other user's letter guess as input", "check if letter is in word", ...)

2. (30) You will receive more specific requirements later, this design is based on your own ideas to start with, so if these following requirements were imposed upon your program and design, then **how might you implement each of the following**:

   a. (05) it needs to store player 1's "secret word",

   b. (05) the "secret word" will only be allowed to contain letters (at least for now),

   c. (05) the program will repeatedly ask for single letters from player 2,

   d. (05) the program will need to show correct letter guesses and the word so far based on those correct guesses,

   e. (05) player two will only get so many guesses (how will you decide this?),

   f. (05) when done, the program will need to **show the secret word, list which letters were missing still, and ask if the players want to play again.**

   Discussion ideas on Discussion Board: what design considerations you think you need to make for each of the above requirements. Are there more major requirements? What are the minor considerations for each of the major considerations?

Remember to submit your report **(in PDF)** and source files **(.cpp files) to TEACH before the end of Sunday**.

Submission Checklist:

makefile

Exercises (20 pts):
    strings.cpp
    name.cpp
    randNum.cpp
    loop.cpp
    randNum2.cpp

Project (40 pts):

    wordGuessDesign.pdf

Report, ***in PDF format***, must address these four sections (30 pts):

    Understanding
    Design
    Testing
    Reflection

The implementation part of the assignment is the .cpp file you submit.