CS 116 Spring 2020 **Lab 2**

Live demo in VM due by March 19, 1:30pm (20 points)
CANVAS DUE DATE for source file(s) and report due by March 19, 11:59pm (80 points)

Purpose

This lab assignment is to provide you with an opportunity to:

- demonstrate your ability to write a C++ program utilizing basic C++ concepts (specific concepts should be listed as part of your report)
- demonstrate understanding of concepts covered, including those related to class

Lab Requirements

This lab assignment requires in-person verification and a lab report. For full credit for this lab, these following files must be submitted via Canvas by the assigned due date, with the live demos completed as stated:

- a lab report which meets the requirements as described below
- your source file(s)
- live demonstration(s) should be completed by the deadline(s) listed above.

Live Demos

During live demo sessions, I may ask for volunteers at first or randomly call names. If at any time you are not ready when your name is called, up to 5 points may be deducted from your score each time. All live demos are required in the VM and are not acceptable in any other IDE.

Description

Write a program that will

a) Ask for the user's name and greet the user.
b) Invite the user to help you train your bug to create secret messages. Give your bug a nice friendly name.
c) If user agrees, prepare your bug to "trace" out at least five 2- or 3-letter words. (There are no visible trails. That's why it's a secret message!)
d) Randomly choose from your list and show the user the gif.
e) Ask the user to guess the word
f) If it is correct, congratulate the user. If it is not correct, provide a message about how the bug needs more training.
g) Thank the user and allow user to repeat or quit. (If repeat, your bug will trace out another randomly chosen word from your list, etc.)
h) Name the main file with your first initial + first 4 letters of your last name + "bug". Since this is a C++ program, the suffix should be .cpp. (For my file, it would be "schenbug.cpp"). Be sure to use programming practices as discussed, and at least 3 meaningful functions outside of main (). You need at least one class object

definition, with appropriate private data elements and public member functions. Use a header file with appropriate content and preprocessing directives. Use one .cpp file for function definitions and one .cpp file for main().

Ask at least 5 non-CS experienced individuals to test your bug. Log down the outcomes and provide the success rates of the messages. Take screenshots of your program during development and testing. These may be handy for your report.

Lab Report requirements

The lab report should have these required sections:

- Purpose
- Planning and organization
- Product
- Pitfalls
- Possible improvements

The lab report should be submitted as a .pdf in Canvas. You may create the report in a document creation software of your choice. Please ensure that your lab report is cleanly formatted and free of distracting grammatical or other errors. These errors will cost you points on clarity. For more specifics, check Canvas for the grading rubric.

Purpose

Provide a few sentences describing the purpose of this assignment, not just the purpose of the program. (You may include a description of what your program does. But this is different than why we are doing this assignment.) Please use your own words. (I don't need to read my own words again.) This should be in a short paragraph form. (A paragraph includes complete sentences, not a collection of phrases that look like a series of text messages.)

Planning and organization

Describe the process you followed in order to plan out and structure this project. The steps here should be specific enough for others to replicate your process. Provide screenshots, other visuals, urls, etc. as needed. You do not need to reproduce the assignment directions, but provide other information to allow another to be successful in this assignment. All visuals should be titled and discussed, i.e. not just stuck on the page with no context. (If partner or team project, each person's contributions are clear.)

This should also be mainly in paragraph form, with lists of steps and visuals as needed. Please use your own words. The description does not need to include every detail, but, as mentioned above, should provide enough information so that your process can be replicated.

Product
Describe your program run(s). Include screenshots for clarity. Discussions should include how the test cases show the capabilities and limitations of your program. The success rate can be discussed here. See the rubric for more details.

Pitfalls
Describe any difficulties and issues you encountered during this assignment. Provide screenshots and other visuals as needed to describe these clearly. Describe how you resolved these issues. If somehow, you didn't have any difficulties, issues, programming errors, or bugs, include an explanation of why this could be the case. See the rubric for more details.

Possible improvements
What could you have done differently to improve your completion of this assignment? Be specific. What could be improved about this assignment overall for you and/or for future students? Be as specific and constructive as possible. See the rubric for more details.

Program file submission requirements
Submit the .cpp, .pdf, and .txt file via Canvas.
To download a file from the VM …

- download directly from localhost access OR
- copy the file to /var/www/html directory. Then it will show on localhost. Click or right-click to download file.

To upload a file to your VM … http://localhost:8080/cgi-bin/file_upload.py