In Class Assignment – Word Jumble

CS244

Due: "In-Class" Check D2L drop box for absolute deadline

Once the dropbox closes, submissions will not be accepted

Overview

Explore and acquaint yourself with Ubuntu Linux, g++, and the basics of C++ programming.
Topics: Multi-Dimensional Arrays

General Objectives:

Explore and learn about:

Arrays and general C++ programming methods

First Step

In Linux in your …/Documents/Programs folder
Create a folder named ***ICA003***

This is necessary as you will compress the ICA003 folder and its contents for submission.

Second Step Check D2L

There may be starter code for this assignment. Likely posted near where this document was found. It may make things easier, but you are not explicitly required to use it.

The 1 file you will need for this assignment MUST be named: WordJumble.cpp

with all your code located in WordJumble.cpp.

All files should have comments with the file name, your name, the last modified date, and a description of the file contents. Each function should be appropriately commented in the header and implementation files. The body of each function may need comments for the less than obvious parts of the code (if any).

Description

- The goal is to
    - Create a puzzle game in which the computer creates a version of a word where the letters are in random order. The player has to guess the word to win the game. If the player is stuck, he or she can ask for a hint
- Follow the starter code as provided on D2L
    - Be sure to rename the file correctly
- Include the files: iostream, string, cstdlib, ctime
- Use the std namespace
- Create an enumerated type named fields with items: WORD, HINT, NUM_FIELDS
    - WORD should equal 0, HINT should equal 1, and NUM_FIELDS should equal 2
- Create a constant integer named NUM_WORDS
    - Set its value to 5
        - After submitting the program you can add your own words and increase the number
- Create a 2D array of constant std::strings named ALLWORDS
    - size should be NUM_WORDS by NUM_FIELDS
        - i.e. 5 by 2
    - Populate the array such that
        - index[0][0] is the word: wall
        - index[0][1] is the hint: Do you feel you are banging your head against something?
        - index[1][0] is the word: glasses
        - index[1][1] is the hint: These might make the answer more clear.
        - index[2][0] is the word: creep
        - index[2][1] is the hint: Going slowly, is it?
        - index[3][0] is the word: persistent
        - index[3][1] is the hint: Keep at it.
        - index[4][0] is the word: jumble
        - index[4][1] is the word: What the game is all about.
- Pick a random word, store it in a variable named choice
- Create a string variable named theWord and set it equal to the selected choice
    - ALLWORDS[choice][WORD]
- Create a second string variable named theHint and set it to the selected word's hint
    - ALLWORDS[choice][HINT]

- Create another string variable named *jumble*
    - Initialize it to theWord
- Set an integer value equal to the length of theWord
- Use a for-loop to mix up the letters in the string named *jumble*.
    - int index1 = (rand() % length; // and likewise for index2
    - swap the characters of jumble at index1 and index 2
        - use: char temp = jumble[index1];
- All is setup now welcome the player to the game
    - Provide directions on how to play
    - Entering the word: quit
        - should end the program
    - Entering the word: hint
        - should provide the player with a hint
    - Output the jumbled word
- Create a string named *guess*, to store the player's guess
- Ask for the first guess and store it in the variable named *guess*.
- Create a while loop that exits if the guess equals theWord or "quit"
- In the while loop
    - Check if guess == "hint"
        - if yes print theHint
    - else
        - print "Nope, that is not it. Try again."
    - Ask the player to enter another guess, store the result in variable: *guess*
- When the loop ends Check if the player quit or solved the jumble
    - Output a message indicating which happened
    - Thank the player
    - Return 0

Tester File

- N/A

makefile

N/A

To Compile at the Command Line Type: g++ WordJumble.cpp

To Run the executable created at the Command Line Type: ./a.out

Suggested Steps to Complete this Assignment

- Keep it simple.
- Use good variable names
- Make sure you put the file name, your name, the due date, and a program description in the comments at the top of the program file
- There is a starter file on D2L (not named as the file to turn in, so it does not get confused with the actual program you are writing)

Grading

10 points possible

Various test inputs/guesses will be used by the grader on multiple runs.
Comments and code formatting will be examined.
Some (but not all) guaranteed ways to lose points include:

Minus 9 if the program fails to compile
Minus 9 if the program fails to run to completion
Minus 9 for any infinite loops
Minus 3 for failure to put name in comments at top of file
Minus 3 for failure to put the due date in the comments at top of the file
Minus 2 for no (or poorly written) program description
Minus 1 for incorrect source filename

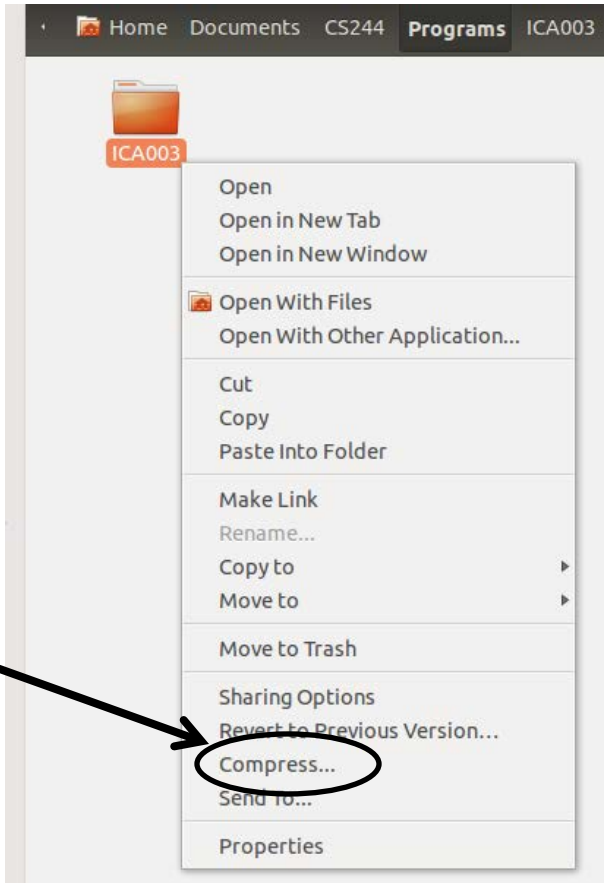
Turn-In Directions

Correctly submitting your work is worth 0 points,
but if not done correctly will likely result in nothing to grade.

Preparation

In Ubuntu Linux browse to your ICA003 folder
Make sure your source code files are in the folder

Right click on the ICA003 folder,
Select Compress…

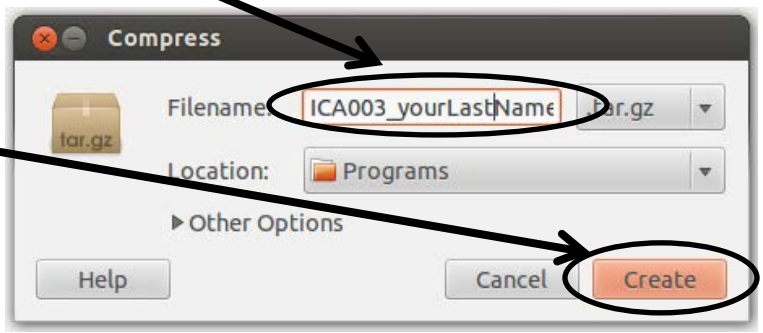


Set the filename to be ICA003_*yourlastname*.tar.gz
where *yourlastname* is your last name

Example: if your last name is
Gollygee then the filename would be
ICA003_Gollygee.tar.gz

Click on the Create Button

This should create the file
named ICA003_ *yourlastname*.tar.gz



Submit

Submit the ICA003_ *yourlastname*.tar.gz file
to the correct course drop box in D2L