**CSCI 1411 – Lab 05 – Functions and Multiple Files**

**Goals:**

* Use of functions
* Use of .h and multiple .cpp files
* Passing parameters by value and reference

**Development Environment:** csegrid (Centos), g++ (all students must use csegrid)

**Skills:** void functions, return functions, function prototypes, function definitions.

Reading: Chap 6

**Deliverables:** 1) This lab with 5 screen shots 2) lastnameFirstLab05.cpp

**Part I – Skills Practice (5 points)**

* Log in to the csegrid. Make a directory called lab05 (remember it is case sensitive). Change directory into **lab05**. Create a file called **Lab05a.cpp** using nano (use your last name and your first initial)

**mkdir lab05  
cd lab05  
nano lab05a.cpp**

* Place the following text in the file (Do not cut and paste. You will learn more by typing it in. If you do cut and paste all of the single and double quotes will be incorrect)

**#include <iostream>**

**#include "Convert.h"**

**using namespace std;**

**int main()**

**{**

**double fahren, celsius;**

**int value1 = 25;**

**cout << "Temperature in Fahrenheit:";**

**cin >> fahren;**

**celcius = fahrenToCel(fahren);**

**cout << fahren << " degrees fahrenheit equates to " << celsius << " degrees Celsius\n";**

**cout << "Before running passByValue value1 = " << value1 << endl;**

**passByValue(value1);**

**cout << "After running passByValue value1 = " << value1 << endl;**

**cout << "Before running passByReference value1 = " << value1 << endl;**

**passByReference(value1);**

**cout << "After running passByReference value1 = " << value1 << endl;**

**return 0;**

**}**

Exit and Save

* Create another file called Convert.h
* **nano Convert.h**

**#ifndef FUNCTIONS\_H**

**#define FUNCTIONS\_H**

**//function prototypes belong in .h file**

**double fahrenToCel(double fahren);**

**void passByValue(int value);**

**void passByReference(int &ref);**

**#endif**

* **Save** the file and exit out.
* Create another file called Convert.cpp

**#include "Convert.h"**

**double fahrenToCel(double fahren)**

**{**

**double result = (fahren - 32) \* (5.0 / 9.0);**

**return result;**

**}**

**void passByValue(int value)**

**{**

**value++;**

**}**

**void passByReference(int &ref)**

**{**

**ref++;**

**}**

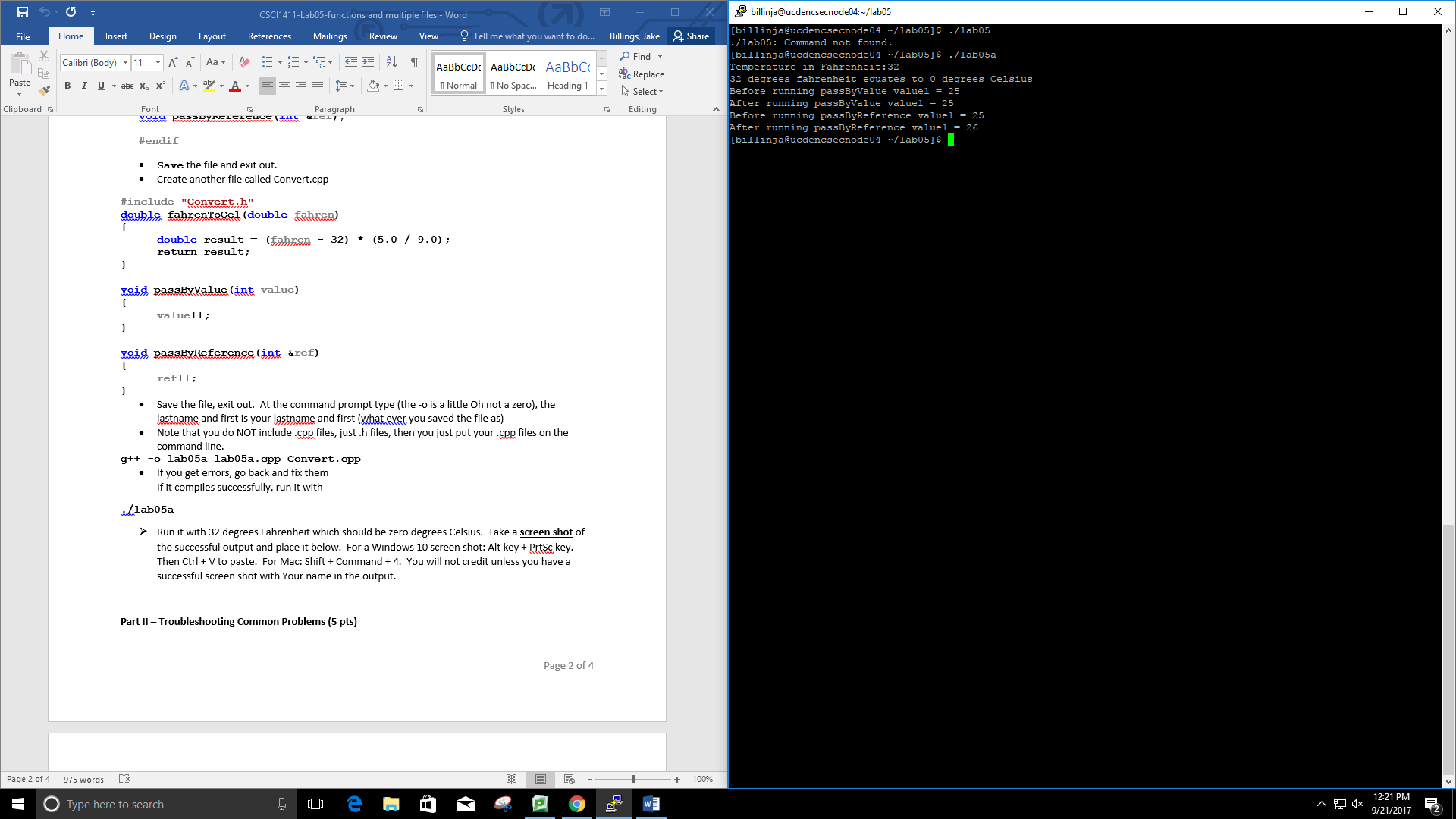
* Save the file, exit out. At the command prompt type (the -o is a little Oh not a zero), the lastname and first is your lastname and first (what ever you saved the file as)
* Note that you do NOT include .cpp files, just .h files, then you just put your .cpp files on the command line.

**g++ -o lab05a lab05a.cpp Convert.cpp**

* If you get errors, go back and fix them  
  If it compiles successfully, run it with

**./lab05a**

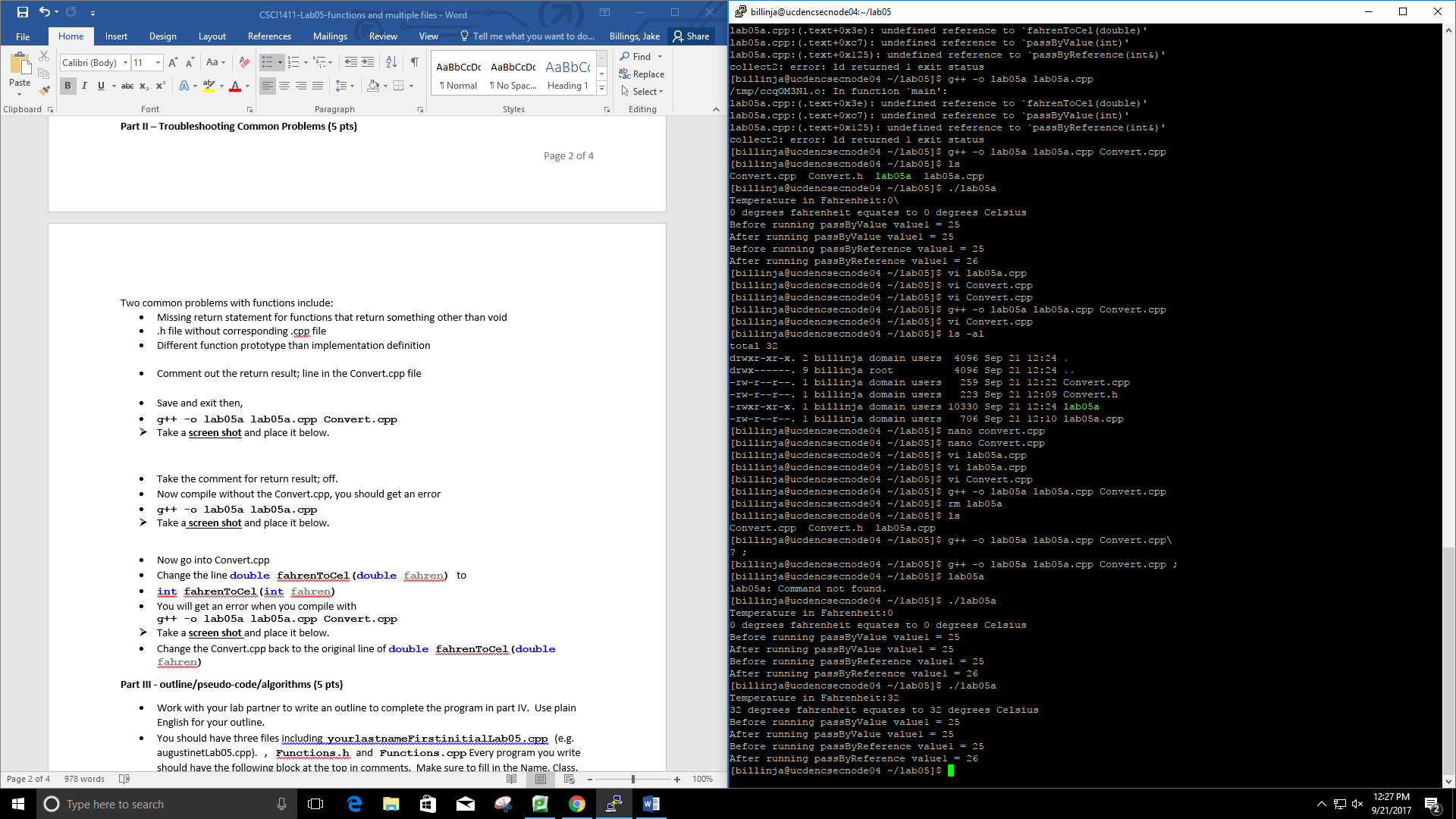
* Run it with 32 degrees Fahrenheit which should be zero degrees Celsius. Take a **screen shot** of the successful output and place it below. For a Windows 10 screen shot: Alt key + PrtSc key. Then Ctrl + V to paste. For Mac: Shift + Command + 4. You will not credit unless you have a successful screen shot with Your name in the output.



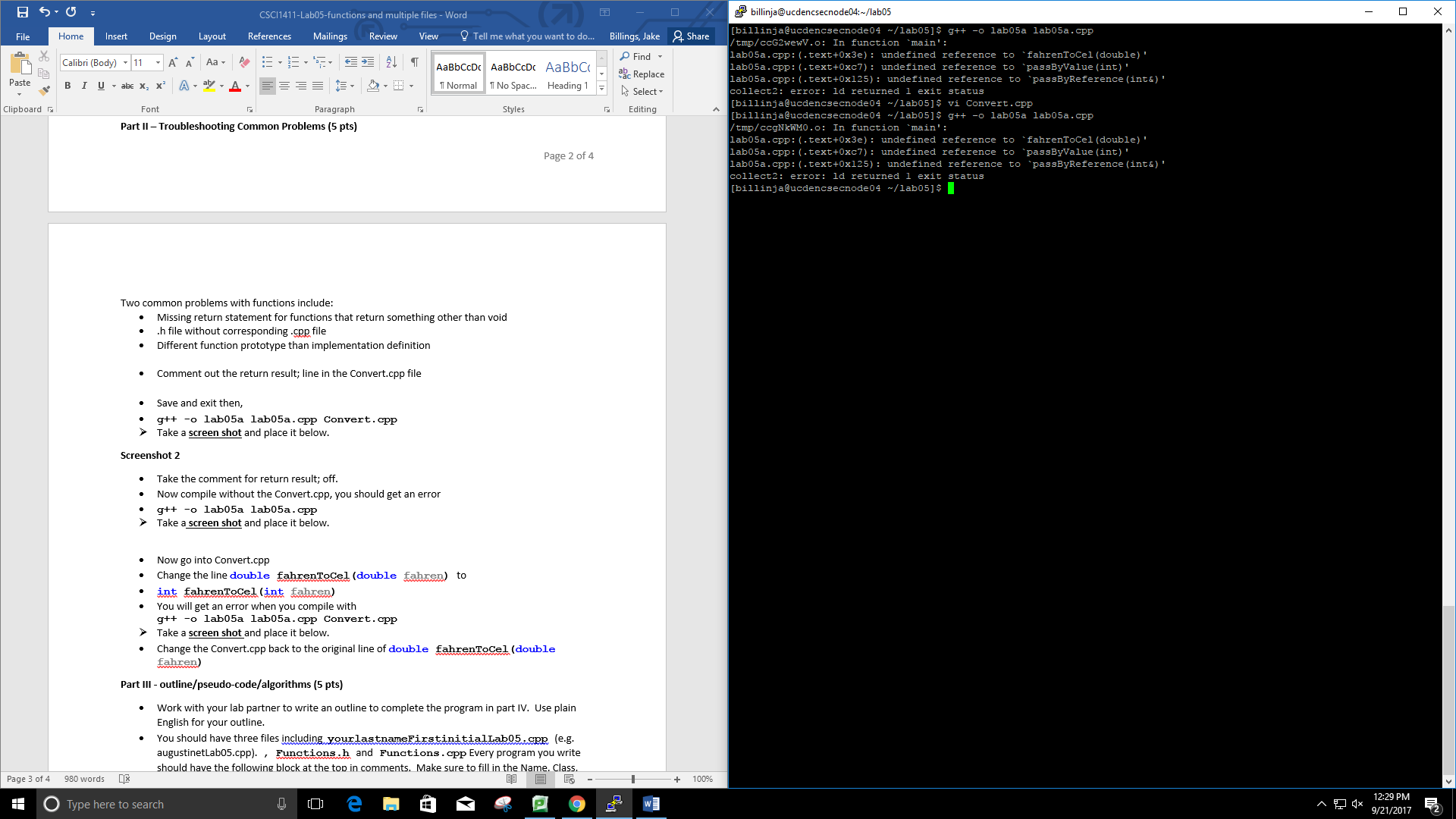
**Part II – Troubleshooting Common Problems (5 pts)**

Two common problems with functions include:

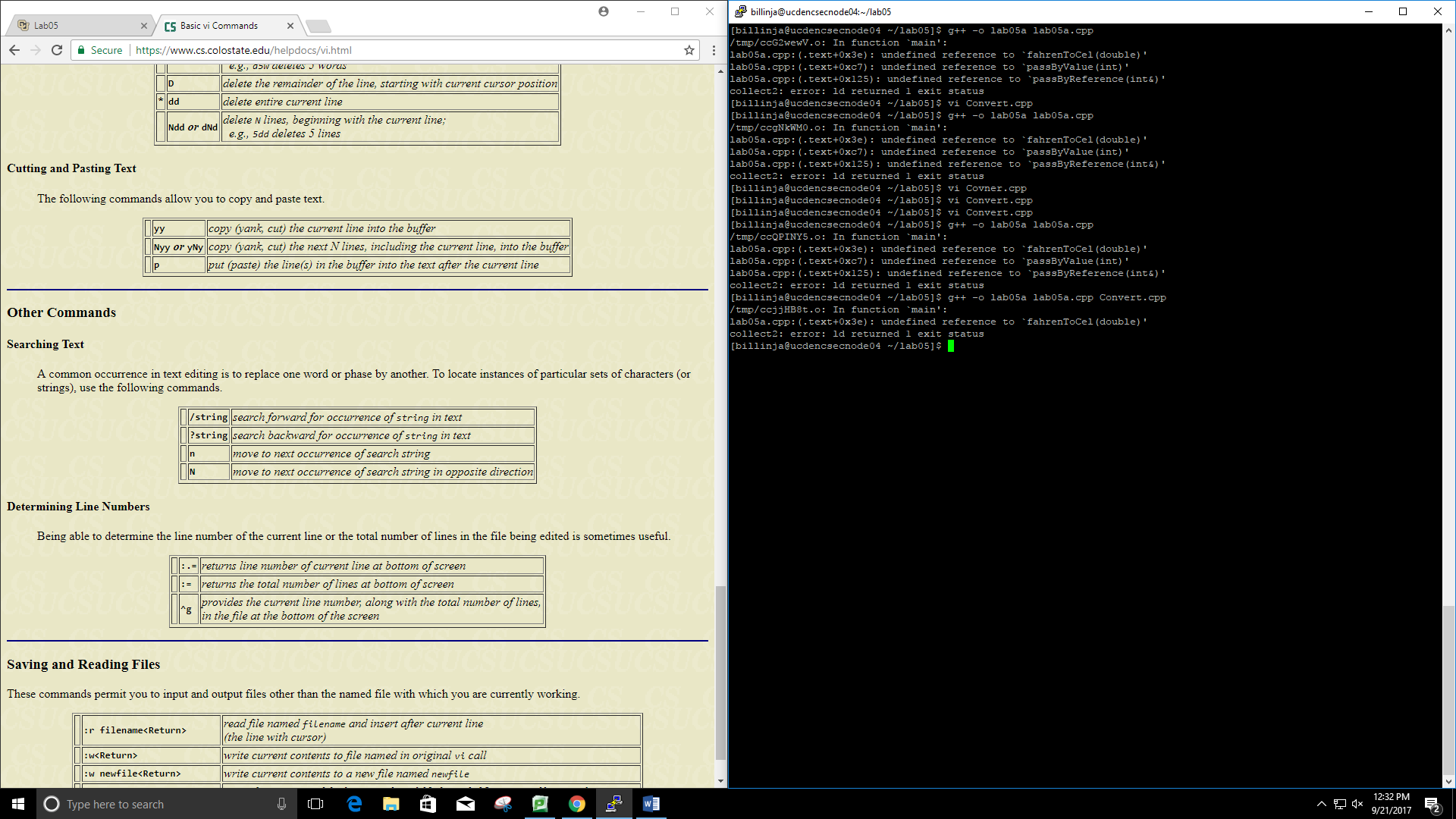
* Missing return statement for functions that return something other than void
* .h file without corresponding .cpp file
* Different function prototype than implementation definition
* Comment out the return result; line in the Convert.cpp file
* Save and exit then,
* **g++ -o lab05a lab05a.cpp Convert.cpp**
* Take a **screen shot** and place it below.



* Take the comment for return result; off.
* Now compile without the Convert.cpp, you should get an error
* **g++ -o lab05a lab05a.cpp**
* Take a **screen shot** and place it below.



* Now go into Convert.cpp
* Change the line **double fahrenToCel(double fahren)** to
* **int fahrenToCel(int fahren)**
* You will get an error when you compile with   
  **g++ -o lab05a lab05a.cpp Convert.cpp**
* Take a **screen shot** and place it below.
* Change the Convert.cpp back to the original line of **double fahrenToCel(double fahren)**

****

**Part III - outline/pseudo-code/algorithms (5 pts)**

* Work with your lab partner to write an outline to complete the program in part IV. Use plain English for your outline.
* You should have three files including **yourlastnameFirstinitialLab05.cpp** (e.g. augustinetLab05.cpp). **, Functions.h** and **Functions.cpp** Every program you write should have the following block at the top in comments. Make sure to fill in the Name, Class, Description and Lab Partner on every file. Ensure your status is accurate.

/\* Name:  
Class: CSCI 1411-00X  
Description: [fill in description]

Lab Partner:  
Status: successfully compiled and run on csegrid [if it doesn’t run or meet all of the requirements, list the actual status!/\*

**Part IV -Dice Roll. (10 pts)**

* Write a function named **rollDice** that simulates the tossing of a multi-sided dice. It should take in parameters that depict how many sides (faces) are on the dice (numbered 1 to the number of sides). When you call the function, it should generate a random number in the range of 1 through the number of dice sides. It should return the number rolled. Then you should have another function called **playGame**. This function take in a parameter that lists the goal of the game. You will alternate roles for each of two players, by calling rollDice (from playGame). The first player to get a score that is equal to that number without going over wins. The function should show that status of each player’s roles. Then it should return 1 or 2 (representing whether player 1 or player 2 won the game. Your main function should call playGame. Then it should print out which player won the game (from main).
* Implementation detail 1: You should place the function prototypes in a file called Functions.h The function prototype has a semi-colon at the end. e.g

**int rollDice (int sides);**

* Implementation detail 2: Your functions.h file should have a “guard” against duplication, e.g.

**#ifndef FUNCTIONS\_H**

**#define FUNCTIONS\_H**

… (function prototypes)

**#endif**

* Implementation detail 3: Your functions.cpp AND your lastnamefirstLab05.cpp should

**#include “Functions.h”.**

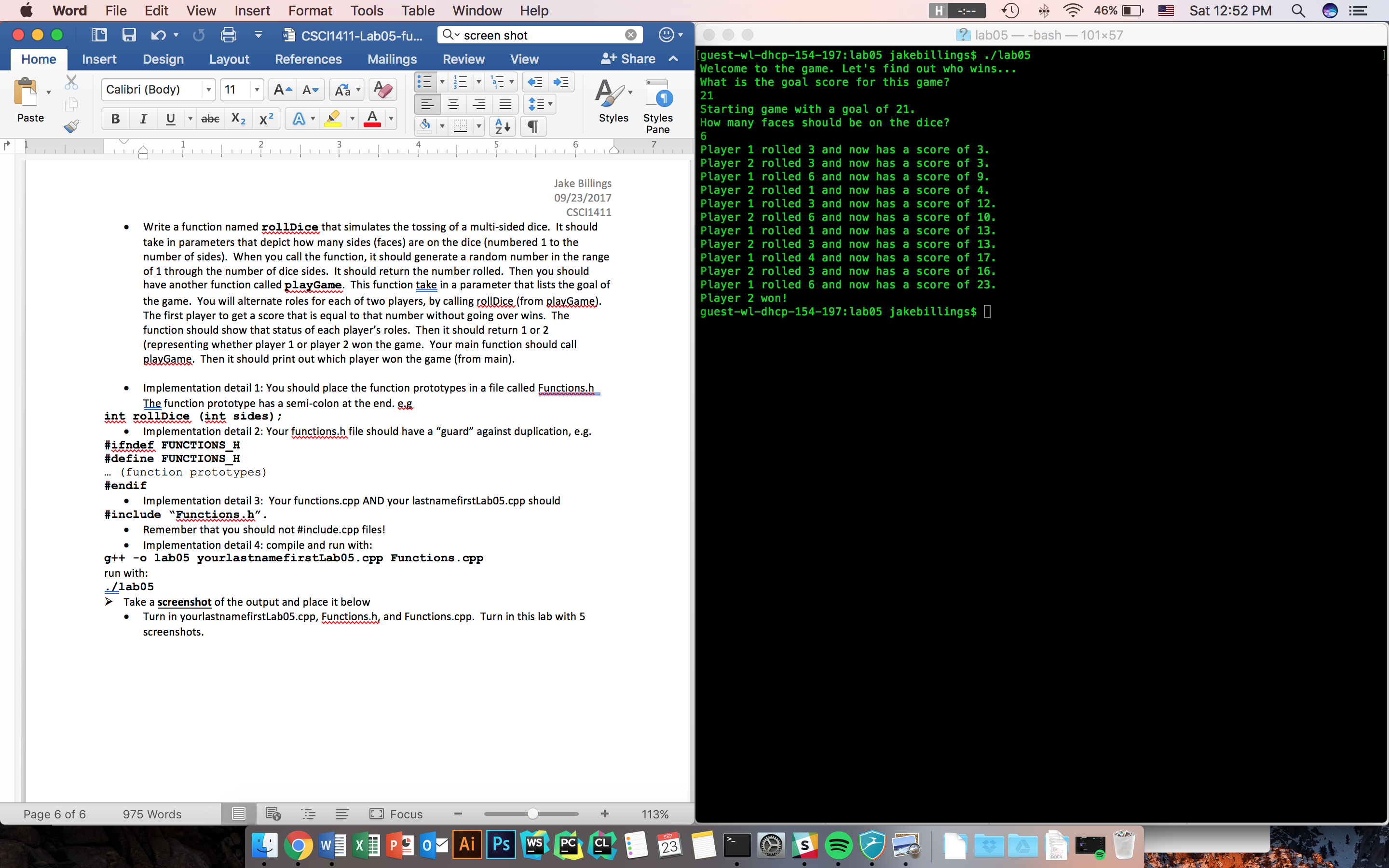
* Remember that you should not #include.cpp files!
* Implementation detail 4: compile and run with:

**g++ -o lab05 yourlastnamefirstLab05.cpp Functions.cpp**

run with:

**./lab05**

* Take a **screenshot** of the output and place it below
* Turn in yourlastnamefirstLab05.cpp, Functions.h, and Functions.cpp. Turn in this lab with 5 screenshots.

****