

# CS 210 - Fundamentals of Programming I

## Fall 2019 – Programming Project 4

### 20 points

**Out: September 19, 2019**

**Due: October 1, 2019 (Tuesday)**

### Problem Statement

The problem statement for this assignment is Programming Project 4 at the end of Chapter 6 on pages 368-370 of the textbook.

### Assignment

Implement a program in C that will repeatedly ask the user for a slope-intercept problem until the user wants to quit. In particular, the program should print a menu for the two choices (two-point or point-slope form), ask the user for a choice, then do the appropriate conversion to slope-intercept form. The program displays both the original form and the slope-intercept form.

### Design Criteria

Your program must conform to the following design criteria in order to earn full credit:

- The program must implement all of the functions given in the problem statement as specified. Reminder: the instructor uses terms "received/passed back parameters" whereas the textbook uses terms "input/output parameters". The textbook also talks about "returning" through output parameters, which is a misnomer, since they are not transmitted back to the caller using a return statement. Again, the instructor uses the phrase "passing back to the caller". Other functions may be used, if you wish.
- The result of function `get_problem` must be validated in the main program (i.e., checked that it is **1 or 2**). If the result is not valid, an error message should be displayed and the program should ask the user if she wishes to continue.
- The program must use a `user_wants_to_continue` kind of function as demonstrated in class and used in the previous programming assignment to control whether the program does another problem.
- The output of the program must conform exactly to the example that follows (user input in **bold**). **Note that this output is slightly different than that shown in the textbook.**

### Coding Notes

- All numbers are displayed with exactly 2 places after the decimal point.
- There are 15 dashes in the two-point form display.
- Display the coordinate, slope, and intercept values as entered or computed in the equation forms. This means that sometimes the display will look like "**4.20x + -3.20**", etc., when it would normally be written "**4.20x - 3.20**", etc.
- Reminder: all lines of output except for prompts end with a single newline. A blank line should be generated by putting a newline at the beginning of the `printf` statement for the following line.

**Example Run (user input in bold)**

SLIPPERY SLOPES

Select the form that you would like to convert to slope-intercept

- 1) Two-point form (you know two points on the line)
- 2) Point-slope form (you know the line's slope and one point)

Enter your choice: **2**

Enter the slope: **4.2**

Enter the x-y coordinates of the point (separated by spaces): **1 1**

Point-slope form

$$y - 1.00 = 4.20(x - 1.00)$$

Slope-intercept form

$$y = 4.20x + -3.20$$

Do you want to do another problem (y/n)? **Y**

Select the form that you would like to convert to slope-intercept

- 1) Two-point form (you know two points on the line)
- 2) Point-slope form (you know the line's slope and one point)

Enter your choice: **4**

Invalid choice!

Do you want to do another problem (y/n)? **y**

Select the form that you would like to convert to slope-intercept

- 1) Two-point form (you know two points on the line)
- 2) Point-slope form (you know the line's slope and one point)

Enter your choice: **1**

Enter the x-y coordinates of the first point (separated by spaces): **4 3**

Enter the x-y coordinates of the second point (separated by spaces): **-2 1**

Two-point form

$$(1.00 - 3.00)$$

m = -----

$$(-2.00 - 4.00)$$

Slope-intercept form

$$y = 0.33x + 1.67$$

Do you want to do another problem (y/n)? **p**

Bad input. Try again.

Do you want to do another problem (y/n)? **n**

That's all!

REMINDER: Your program must compile for it to be graded. Submissions that do not compile will be returned for resubmission and assessed a late penalty. Submissions that do not substantially work also will be returned for resubmission and assessed a late penalty.

Follow the program documentation guidelines in the [C Programming Style Guideline](#) handout. As stated in the syllabus, part of the grade on a programming assignment depends on how well you adhere to the guidelines. The grader will look at your code and grade it according to the guidelines.

### **What to Submit**

Electronically submit a zipfile containing `main.c` (only) as explained in the handout [Submission Instructions for CS 210](#). The submission system will begin accepting assignments no earlier than the evening of Tuesday, September 24 (i.e., after Practical Exam 1).