

CS 118 — Programming Fundamentals

Assignment #3

Due Date: Sunday, September 22nd at 11:55pm on Google Classroom.

Instructions: Assignments are to be done individually. **No late assignments will be accepted.** You must complete this assignment by yourself. You cannot work with anyone else in the class or with someone outside of the class. The code you write must be your own. You are encouraged to get help from the instructional staff. You may post general questions on Piazza. Do not post more than one line of code when using Piazza.

You must **submit a single zip file** containing your code and documentation on Google Classroom named `<your_student_id>.zip` where `<your_student_id>` is something like `i19-XXXX`. This means that you must submit only **one file named `i19-XXXX.zip` containing only your source files**. Each file that you submit **must contain your name, student-id, and assignment#** on top of the file in comments. Your submission must NOT contain multiple `main()` functions, otherwise it will not compile for grading. Test your program on a lab machine before submission.

Follow the instructions. Assignments not following the instructions will be awarded zero points.

Assignment Statement: Rock, Paper, Scissors is a two-player game played as a means to help make a choice or put an end to a disagreement. The idea is that both players have an equal chance of winning, making the game random but fair. The game of Rock, Paper, Scissors is played like this: at the same time, two players display one of three symbols: a rock, paper, or scissors. A rock beats scissors breaking it, scissors beat paper by cutting it, and paper beats rock by covering it.

In this assignment you will write a program to allow a human to play a game of Rock, Paper, Scissors versus the computer. The sample programs (`rps1.txt`, `rps2.txt`, and `rps3.txt`) show 3 runs from my solution to this problem using the default computer player. You must use the given program template so the choices are unpredictable. This is necessary so that we can grade your program.

Given the same human name and computer choices, your program must match this output exactly. Even minor differences in characters will cause you to fail grading tests and lose points.

The program:

- Describes the game to the user (done for you!)
- Asks the user for their name (done for you!)
- Plays three rounds of the game
- In each round of the game
 - Asks the user for their choice
 - Have the computer make a random choice
 - Prints out each player's choice
 - Prints the results of the round
- After playing the three rounds, displays the number of rounds the user won, the number of rounds the computer won, and the number of rounds that were a draw
- Declares who the better player was based on the number of wins

This is not an easy program, mostly due to the size of the program. The individual steps are not too difficult, but there are many steps. The program description above gives you a rough idea of how to break the program up into parts.

Have a high level structure and then implement parts of that structure (the individual functions) one at a time, testing to make sure they work before going on. You may have to write some testing code that will not be part of the final program. Do not write the whole program in main and then try and break it up into functions.

Here are some tips on the various parts of the program.

1. The `main()` function.

The main function initializes a random selection for the computer. Do not change this code. Feel free to share examples of your output on Piazza.

In the main program calls `scanf()` to ask the user for his/her **name**. You may pass name as a parameter to your functions where you need it. The main function should not have a lot of statements, instead it shall call other functions.

2. Playing the rounds of the game given our current programming tools this will be the largest and most complex function. It is in turn broken down into several parts. You will need some local variables in this function.
3. Ask the user what their choice is. The user will enter an integer as their choice. If the user enters an invalid input the program should print `Invalid choice. You lose.` and count it as win for the Computer.
4. Have the computer make a random choice. To do this call the `getComputerChoice()` function written in `rps.h` which is included in `rps.c`. You do not need to modify this file.
5. Print out each player's choice. You will find it useful to have a function that is passed an `int` parameter and prints the correct string for that `int`. In this program 1 represents "Rock", 2 represents "Paper", and 3 represents "Scissors".
6. Print the results of that round. This is the most algorithmically difficult part of the assignment because there are nine possible outcomes and using the programming tools we have as of now, it is difficult to remove redundancy. However, try to make your program as concise as possible.

The nine possible outcomes are

Computer Choice	Human Choice	Result
Rock	Rock	Draw
Rock	Paper	Human Wins
Rock	Scissors	Computer Wins
Paper	Rock	Computer Wins
Paper	Paper	Draw
Paper	Scissors	Human Wins
Scissors	Rock	Human Wins
Scissors	Paper	Computer Wins
Scissors	Scissors	Draw

You must follow the format as shown in the sample output.

7. After playing the three rounds, display how many times the user won, how many times the computer won, and how many draws occurred. The function that runs the rounds shall call a function to display this information.
8. Finally, declare who the better player was based on the number of wins. This can be part of the results function but will require some conditional execution with conditional statements.

Style issues. We will grade program hygiene as well as correctness. Did you provide a good structure to the program using functions? Did you minimize the scope of variables to the smallest necessary? Did you use meaningful identifiers? Did you provide consistent tabbing and spacing for code inside functions and if statements? Did you provide comments for your functions?

Honor Policy

This assignment is a individual learning opportunity that will be evaluated based on your ability to think independently, work through a problem in a logical manner solve the problems on your own. You may however discuss verbally or via email the general nature of the conceptual problem to be solved with your classmates or the course instructor, but you are to complete the actual assignment without resorting to help from any other person or other resources that are not authorized as part of this course. If in doubt, ask the course instructor. You may not use the Internet to search for solutions to the problem.