San José State University
Department of Computer Engineering

CMPE 180-92

Data Structures and Algorithms in C++

Spring 2017

Instructor: Ron Mak

Assignment #2

Assigned:	Thursday, February 2
Due:	Thursday, February 9 at 5:30 PM
CodeCheck:	http://codecheck.it/codecheck/files/1702020803f5ez0x3oehy26y9z3w82ir61p
Canvas:	Assignment 2. Rock Paper Scissors
Points:	100

Rock Paper Scissors

This assignment will give you practice decomposing a program with functions. Write a C++ program that plays the game of Rock Paper Scissors with a human player (you).

At the above CodeCheck URL, complete the program **rps.cpp** in CodeCheck's edit box, and then press the "Submit" button. CodeCheck will compile and run your program. You can type into CodeCheck's edit box directly, or you can first edit and test your program in the Eclipse or NetBeans IDE, and then cut and paste it into CodeCheck.

The green sheet (syllabus) describes how to set up your programming environment.

Academic integrity

You may study together and discuss the assignments, but what you turn in must be your individual work. Assignment submissions will be checked for plagiarism using Moss (http://theory.stanford.edu/~aiken/moss/). **Copying another student's program or sharing your program is a violation of academic integrity.** Moss is not fooled by renaming variables, reformatting source code, or re-ordering functions.

Violators of academic integrity will suffer severe sanctions, including academic probation. Students who are on academic probation are not eligible for work as instructional assistants in the university or for internships at local companies.

Rules of the game

Each game has two players and consists of several rounds. During each round, each player simultaneously chooses rock, paper, or scissors. The winner of each round is determined as follows:

- Rock can break scissors, so rock wins over scissors.
- Paper can cover rock, so paper wins over rock.
- Scissors can cut paper, so scissors win over paper.
- It's a tie if both players chose the same thing.

For this assignment, a game consists of 20 rounds. (Make 20 a named constant.)

At the start of each round, your program should prompt you to enter your choice. It should accept the letter **`r`** or **`R`** for rock, **`p`** or **`P`** for paper, or **`s`** or **`S`** for scissors. If you entered an incorrect character, your program should print an error message and prompt again for your choice.

Your program should randomly choose rock, paper, or scissors for the computer. It should then determine who is the winner of the round, or if it's a tie.

The game concludes after all the rounds are completed. Your program should print the number of human player wins, the number of computer wins, and the number of ties.

Program design

Your program must be decomposed into functions. You must have at least functions for the following tasks:

- Get the human player's (i.e., your) choice.
- Randomly generate the computer's choice.
- Determine who is the winner.
- A function named **`record_win`** to record the winner of a round.

Function **`record_win`** must have as parameters the winner of the round just completed, the current number of human player wins, the current number of computer wins, and the current number of ties. The function should update the appropriate parameter value.

Your program may have other functions. It may have no global variables.

Choose descriptive variable and function names. Include meaningful comments in your code, but don't over-comment. Include your name in a comment at the top of your program. Follow the coding style (formatting, braces, indentation, etc.) of the Savitch textbook. You must have all your function declarations before the main and your function definitions after the main.

CodeCheck will play as the human player

When you submit your program into CodeCheck, CodeCheck will play the role of the human player. It will play 20 rounds and make a choice (possibly an illegal one) for each round.

Submission into Canvas

When you're satisfied with your program in CodeCheck, click the "Download" link at the very bottom of the Report screen to download a signed zip file of your solution. Submit this zip file into Canvas. You can submit as many times as you want until the deadline, and the number of submissions will not affect your score. Only your last submission will be graded. Due to the random numbers, CodeCheck will not compare your output. Also, because CodeCheck feeds choices (as the human player) to your program, the output may look different from the output you get when you play interactively with your program. What the "human" types might not be displayed**. Ignore the 0 score at the end.**

Submit the signed zip file from CodeCheck into Canvas:
Assignment 2. Rock Paper Scissors.

Note: You must submit the signed zip file that you download from CodeCheck, or your submission will not be graded. Do not rename the zip file.

Sample interactive output

Output when the game is played interactively with a real human.

```
Round 1
  Your choice? r
  You chose ROCK.
  The computer chose SCISSORS.
  The winner is you.

Round 2
  Your choice? R
  You chose ROCK.
  The computer chose PAPER.
  The winner is the computer.

Round 3
  Your choice? x
  *** ERROR: Valid choices are R, P, or S
  Your choice? p
  You chose PAPER.
  The computer chose ROCK.
  The winner is you.

Round 4
  Your choice? s
  You chose SCISSORS.
  The computer chose PAPER.
  The winner is you.

...

Round 20
  Your choice? p
  You chose PAPER.
  The computer chose SCISSORS.
  The winner is the computer.

Summary
-------
   Human wins: 7
Computer wins: 8
         Ties: 5
```

Rubric

Your program will be graded according to these criteria:

Criteria	Maximum points
Correct program logic. • Each round played correctly. • Appropriate output per round.	**30** • 20 • 10
Program decomposition • At least the four functions described above. • Function **`record_win`** written well.	**40** • 20 • 20
Good program style • Descriptive variable and function names. • Meaningful comments. • Function declarations before the main, function definitions after the main. • Follows the coding style (formatting, braces, indentation, etc.) of the Savitch textbook.	**30** • 10 • 5 • 5 • 10