

CSCI 200 - Fall 2023

Foundational Programming Concepts & Design

A1 - Rock Paper Scissor Throw Down!



- This assignment is due by Thursday, September 07, 2023, 11:59 PM.←
- As with all assignments, this must be an individual effort and cannot be pair programmed. Any debugging assistance must follow the course collaboration policy and be cited in the comment header block for the assignment.←
- Do not forget to complete the following labs with this set: L1A, L1B ←
- Do not forget to complete zyBooks Assignment 1 for this set.←

Jump To: [Rubric Submission](#)

Assignment Code Starter

Make sure you have the appropriate comment header block at the top of every assignment from this point forward. The header block should include the following information at a minimum.

```

/* CSCI 200: Assignment 1: A1 – Rock Paper Scissor Throw Down!
 *
 * Author: XXXX (INSERT_NAME)
 * Resources used (Office Hours, Tutoring, Other Students, etc & in what capacity):
 *    // list here any outside assistance you used/received while following the
 *    // CS@Mines Collaboration Policy and the Mines Academic Code of Honor
 *
 * XXXXXXXX (MORE_COMPLETE_DESCRIPTION_HERE)
 */
// The include section adds extra definitions from the C++ standard library.
#include <iostream> // For cin, cout, etc.
```

RMP-H

```
// We will (most of the time) use the standard library namespace in our programs.
using namespace std;

// Define any constants below this comment.

// Must have a function named "main", which is the starting point of a C++ program.
int main() {

    /***** INSERT YOUR CODE BELOW HERE *****/

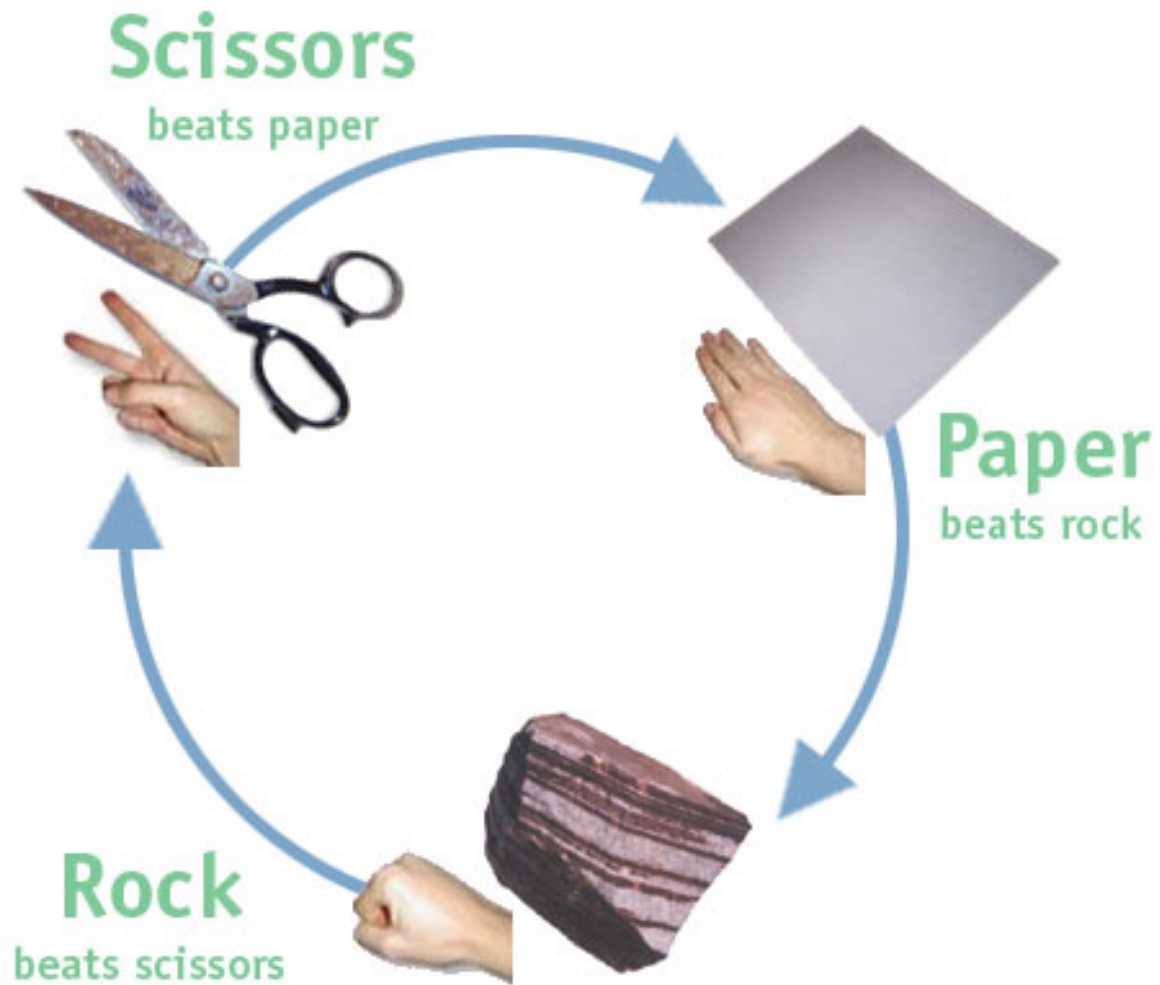
    cout << "Hello world!" << endl; // print Hello world! to the screen

    /***** INSERT YOUR CODE ABOVE HERE *****/

    return 0; // signals the operating system that our program ended OK.
}
```

Rock Paper Scissors

Most of you have likely played the classic game **Rock, Paper, Scissors**. Believe it or not, a **hardcore world** of Rock, Paper, Scissors has existed, even in **Denver**. It's a bit frightening.... So, before you start thinking how Rock, Paper, Scissors (or RPS for those in the know) is a **kid's** game, think again. This is serious stuff; go ahead and TRY to beat **the computer**. (But, if anyone asks, your C++ assignment this week is to implement a simple interactive rule-based system that addresses a theoretical **Decision Problem**.)



Your goal is to ultimately implement a one-player version of **Rock, Paper, Scissors** against the computer.

The User's Choice

The first step is to ask the Human Player what they choose and then repeat back their choice. The Human Player can enter either **R** or **r** for rock, **P** or **p** for paper, and **S** or **s** for scissors. Instead of printing back whatever the Human entered, we want to display the full word that corresponds to the single letter entered. Here is an example interaction for this part of the program:

```
Welcome one and all to a round of Rock, Paper, Scissors! (Enter P, R or S)
Player: R

Player choose Rock
```

Here is another example:

```
Welcome one and all to a round of Rock, Paper, Scissors! (Enter P, R or S)
Player: p

Player choose Paper
```

Note: We will assume a smart user that follows directions and will enter a valid value.

The Computer's Choice

Now, we must randomly decide what the Computer chooses. To do so, we will randomly generate a number for the computer. The computer has three possible choices and we will represent these three choices by the numbers 0, 1, and 2. Properly use the `rand()` function to generate a random number in the range [0, 2]. Instead of displaying 0, 1, or 2, your program should instead print Rock, Paper, or Scissors respectively. At this point, we now will know what each player has thrown:

```
Welcome one and all to a round of Rock, Paper, Scissors! (Enter P, R or S)
Player one: s

Player choose Scissors
Computer choose Scissors
```

```
Welcome one and all to a round of Rock, Paper, Scissors! (Enter P, R or S)
Player one: S

Player choose Scissors
Computer choose Rock
```

```
Welcome one and all to a round of Rock, Paper, Scissors! (Enter P, R or S)
Player one: r

Player choose Rock
Computer choose Paper
```

Every time you run your program, the computer should have a different value. Once your random number is working properly, it's time to make sense of both players' choices and move on.

Declare a Winner

The final step of our game is to determine who actually won. The **handy chart above** will be your guide.

Add a final line of output that prints a line following this pattern:

```
X beats Y. Z wins!
```

Where X and Y are one of "rock" or "scissors" or "paper" and Z is either "Human" or "Computer". Be sure to handle ties appropriately too. And now, we have a fully functioning Rock, Paper, Scissors game! Great job!

```
Welcome one and all to a round of Rock, Paper, Scissors! (Enter P, R or S)
Player one: R

Player choose Rock
Computer choose Paper

Paper beats Rock.
Computer wins!
```

Let's Play Two!

Now we want to modify our game so the user can continue to play another game if they choose.

As the user continues to play, keep track of how many games the user won, lost, and tied. When the user stops playing, print out a nice message and how many games were won, lost, and tied.

A sample run of the program is shown below:

```
Welcome one and all to a round of Rock, Paper, Scissors! (Enter P, R or S)
Player one: R

Player choose Rock
Computer choose Paper

Paper beats rock.
Computer wins!

Do you want to play again? (Y/N) Y

Welcome one and all to a round of Rock, Paper, Scissors! (Enter P, R or S)
Player one: R
```

```
Player choose Rock
Computer choose Paper

Paper beats rock.
Computer wins!

Do you want to play again? (Y/N) y

Welcome one and all to a round of Rock, Paper, Scissors! (Enter P, R or S)
Player one: R

Player choose Rock
Computer choose Scissors

Rock beats scissors.
Player wins!

Do you want to play again? (Y/N) N

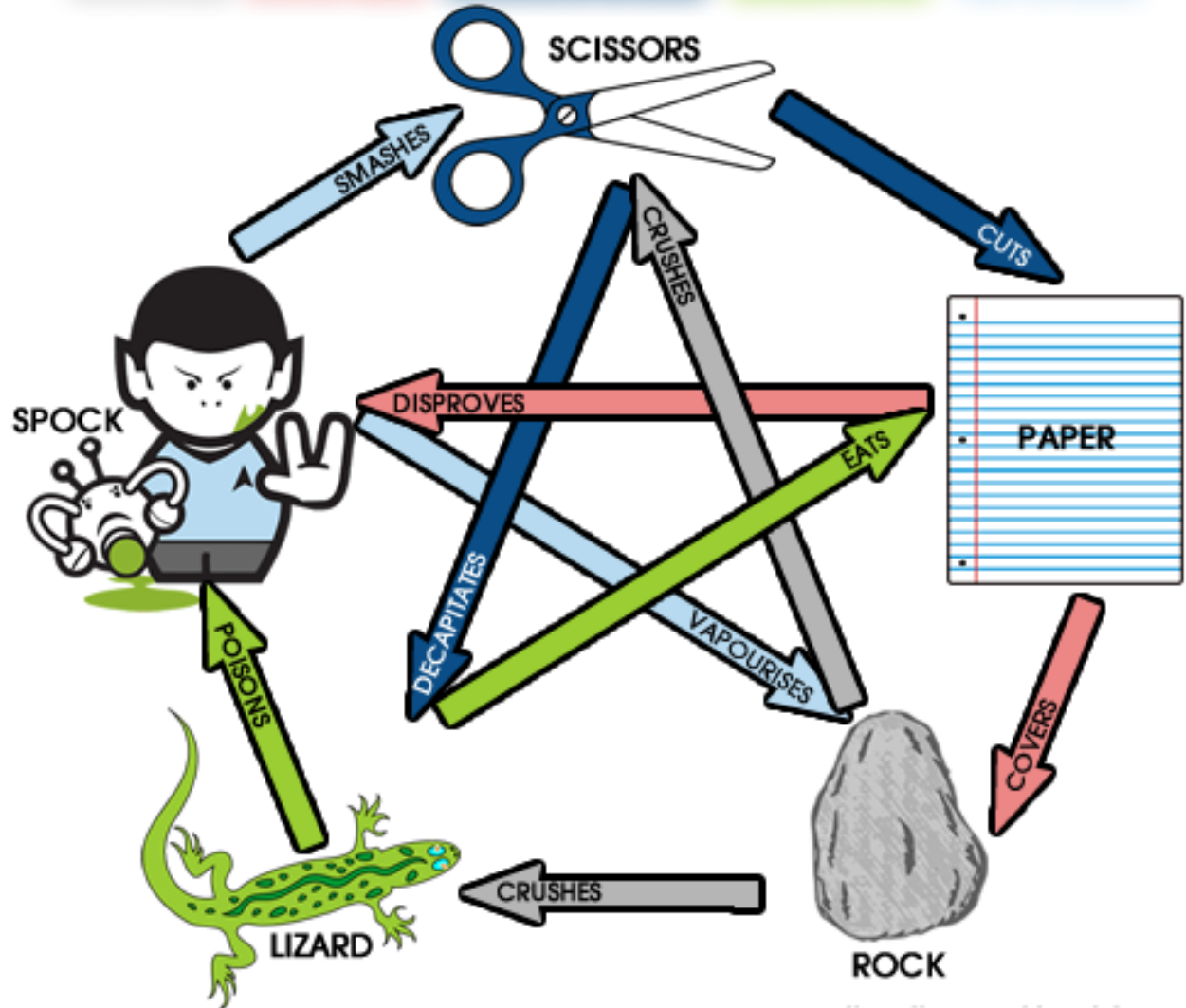
Thanks for playing!
You won 1 game(s), lost 2 game(s), and tied 0 game(s).
```

Congrats! We've now fully finished our Rock, Paper, Scissors game. See how many games you can play in a row without losing to the computer!

Extra Credit! Rock, Paper, Scissors, Lizard, Spock

For extra credit, expand your program to play a game of **Rock, Paper, Scissors, Lizard, Spock**. Feel free to **practice a few games** to fully understand what beats what. Below is the logic. You will notice there are more ways to win or lose and a smaller possibility to tie.

HOW TO PLAY ROCK, PAPER, SCISSORS, LIZARD, SPOCK



Grading Rubric

Your submission will be graded according to the following rubric:

Points	Requirement Description
0.5	Submitted correctly by Thursday, September 07, 2023, 11:59 PM
0.5	Project builds without errors nor warnings.
2.0	Best Practices and Style Guide followed.
0.5	Program follows specified user I/O flow.

0.5	Public and private tests successfully passed.
2.0	Fully meets specifications.
6.00	Total Points

Extra Credit Points	Requirement Description
+0.5	Game expanded to Rock-Paper-Scissors-Lizard-Spock

Submission

Always, **always**, **ALWAYS** update the header comments at the top of your main.cpp file. And if you ever get stuck, remember that there is LOTS of **help** available.

Zip together your `main.cpp`, `Makefile` files and name the zip file `A1.zip`. Upload this zip file to Canvas under A1.

- **This assignment is due by Thursday, September 07, 2023, 11:59 PM.**←
- **As with all assignments, this must be an individual effort and cannot be pair programmed. Any debugging assistance must follow the course collaboration policy and be cited in the comment header block for the assignment.**←
- **Do not forget to complete the following labs with this set: L1A, L1B** ←
- **Do not forget to complete zyBooks Assignment 1 for this set.**←

Last Updated: 08/22/23 12:59

Any questions, comments, corrections, or request for use please contact jpaone {at} mines {dot} edu.

Copyright © 2022-2023 Jeffrey R. Paone



CS@Mines



[\[Jump to Top\]](#) [\[Site Map\]](#)