C2 Pipeline
Innovation and Curiosity Center
Coding Challenge 2 Walk Through

This coding challenge was written by Mikayla Scott under the C2 Pipeline Innovation and Curiosity Center through Wayne State University. This challenge has students work to create a Rock, Paper, Scissors program. The challenge prompt is listed below. For more information on the STEM Challenges produced by C2 Pipeline visit https://c2pipeline.wayne.edu/stem-lab. The YouTube video that accompanies this lesson can be found at https://www.youtube.com/watch?v=3vkQWtErAgc&t=1s

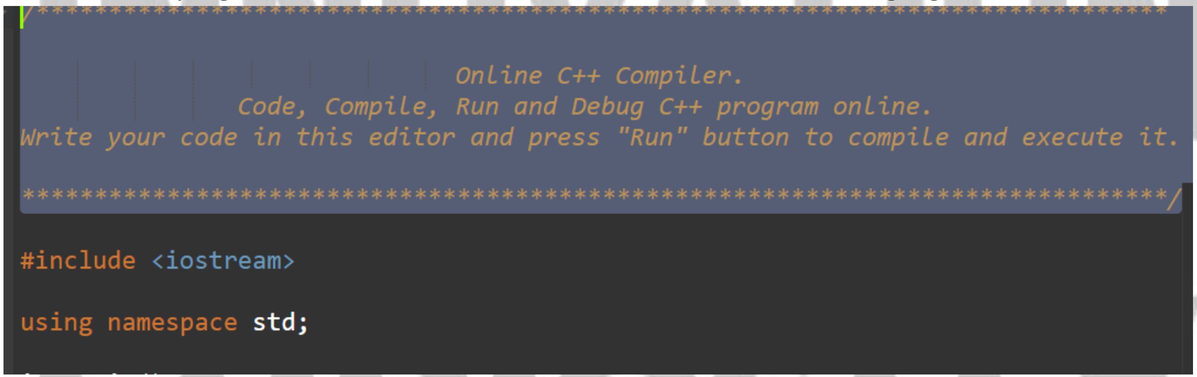
+-

“Code a rock, paper, scissors game where players play against the CPU. The program should include logic operators, if statements and the user of the rand() function. The results should print the winner of the match!”

+-

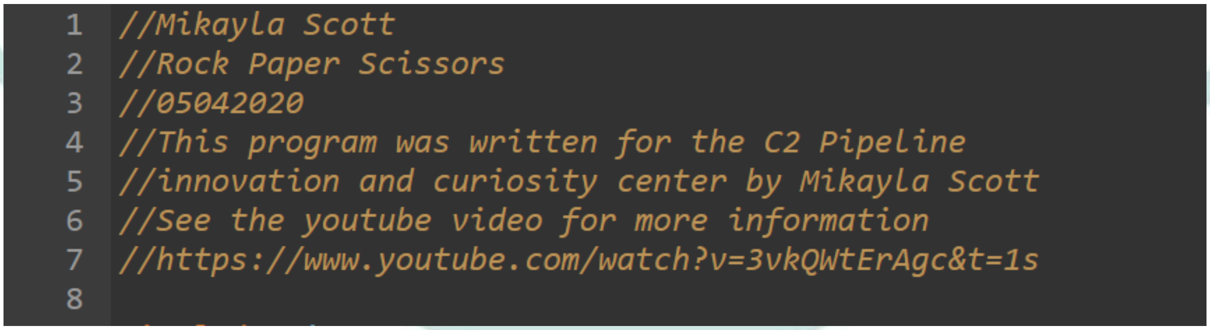
Instructions:

1. **Find an online compiler.** The compiler that is used in the video can be found at the link below: https://www.onlinegdb.com/online_c++_compiler
2. Once the compiler launches, **delete the pre-coded header** from the .cpp file. A header is a block of comments that tell other programmers’ information about the code. The header is highlighted.



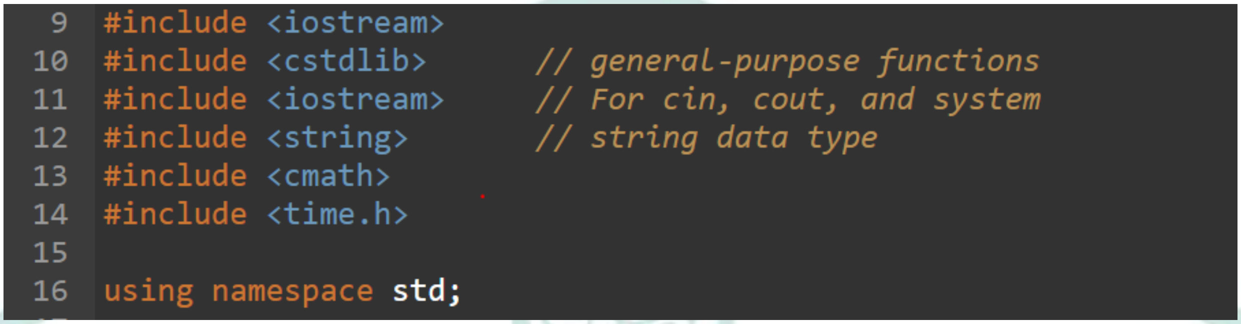
```
/******************************************************************************

                              Online C++ Compiler.
               Code, Compile, Run and Debug C++ program online.
Write your code in this editor and press "Run" button to compile and execute it.

*******************************************************************************/

#include <iostream>

using namespace std;
```

3. Now **define a header** by writing two backslashes **//.** The headers should include, the programmers name, the date it was created, and what the program does. See completed header below.
Tip: Remember comments do not show up in the execution of the program.



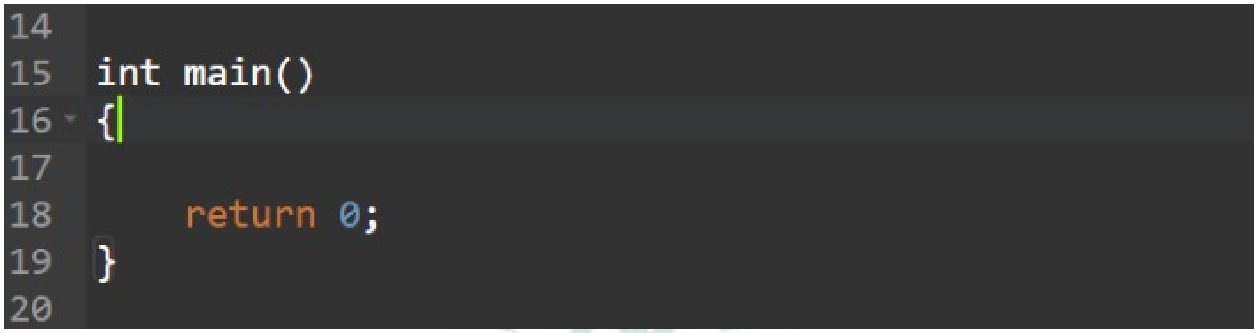
```
//Mikayla Scott
//Rock Paper Scissors
//05042020
//This program was written for the C2 Pipeline
//innovation and curiosity center by Mikayla Scott
//See the youtube video for more information
//https://www.youtube.com/watch?v=3vkQWtErAgc&t=1s

```

4. **Include the following libraries** in your program underneath the header.
Tip: Libraries are a collection of precompiled routines that a program can use. This includes functions, variables and more.



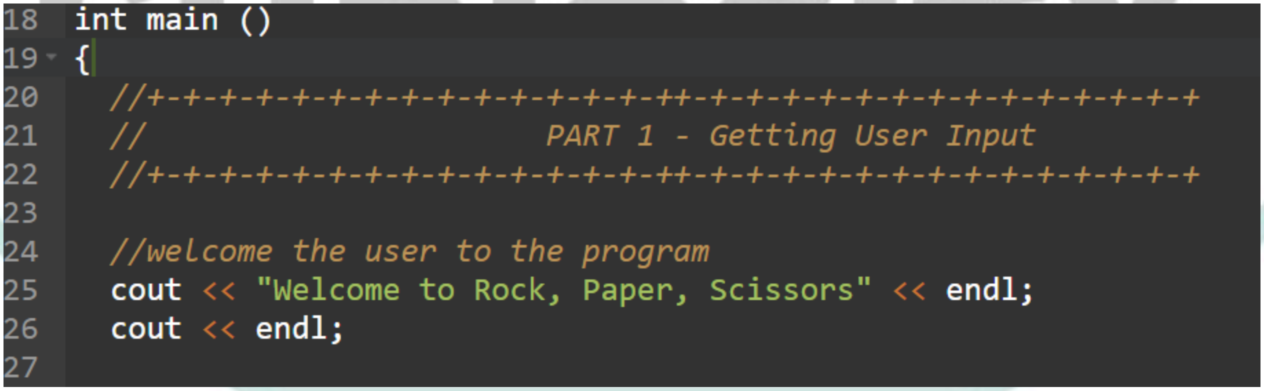
```
#include <iostream>
#include <cstdlib>      // general-purpose functions
#include <iostream>     // For cin, cout, and system
#include <string>       // string data type
#include <cmath>
#include <time.h>

using namespace std;
```

5. **Delete** the line of code in main that says **cout << ”hello world”;** cout stands for characters out, so if you don’t delete this line when you finally run the code it will print hello world to the screen.
Tip: Make sure to leave the return statement, main(), and the { } in the same places. Only delete cout << “hello world”;



```

int main()
{

    return 0;
}

```

6. Time to start coding! First send a welcome line to the user.



```
int main ()
{
    //+-+-+-+-+-+-+-+-+-+-+-+-+-++-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    //                      PART 1 - Getting User Input
    //+-+-+-+-+-+-+-+-+-+-+-+-+-++-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

    //welcome the user to the program
    cout << "Welcome to Rock, Paper, Scissors" << endl;
    cout << endl;

```

7. Now that the user has been welcomed to the program, prompt the user to input either rock, paper or scissors. To do this declare a string variable using the following syntax:

```
string varName;
```

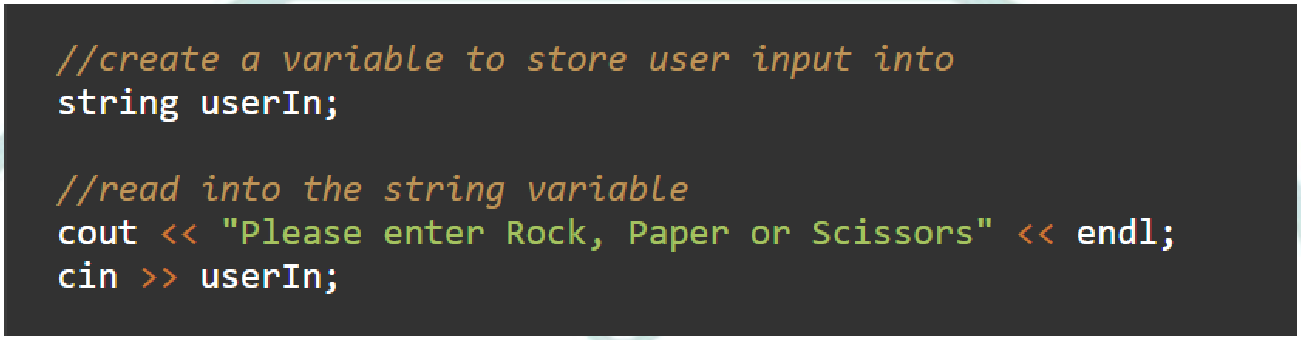
Once the string with your name is declared, send the user a message and ask them to choose rock paper or scissors**.**

```
cout << “Please enter Rock, Paper or Scissors” << endl;
```

Now take characters in from the user and store them in the variable that was declared in the first step.

```
        cin >> varName;
```

see image of all the steps combined below:

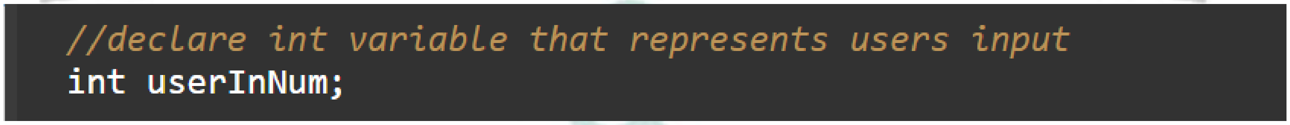


```
//create a variable to store user input into
string userIn;

//read into the string variable
cout << "Please enter Rock, Paper or Scissors" << endl;
cin >> userIn;
```

8. **Declare an integer variable** using the syntax below:

```
        int varName;
```

Tip: This variable will be used to store the number that corresponds to either rock paper or scissors. We have to take the string that is input by the user and correspond that with a number because the rand() fuction (random number generator) that we will be using later on will produce an integer value.



```
//declare int variable that represents users input
int userInNum;
```

9. Associate each of the inputs with 1, 2 or 3.

Rock = 1 *Paper = 2* *Scissors = 3*

To associate each one of these possible inputs with a number, **use an if statement**.

Tip: An if statement accounts for all the cases that a user could enter. This is where the logic statement comes into play. Use the syntax below for your first if statement:

```
        if (condition)
        { //code }
        else if(condition)
        { //Code }
        else
        { //code }
```

The first condition in the if statement comes from what the user entered, if the user entered “Rock” **OR** “Paper” **OR** “Scissors” then we want to take that input and associate it with a number. Keep in mind that the **OR** is going to be the logic operator that we will use in this line of code.

Tip: The logic operator or is coded using this symbol ||

Now the code inside the first if statement is going to be surrounded by curly braces, so make sure that everything is **enclosed within these { }.**

Inside those curly braces, **use another if statement** to break the three options into their own respective cases.

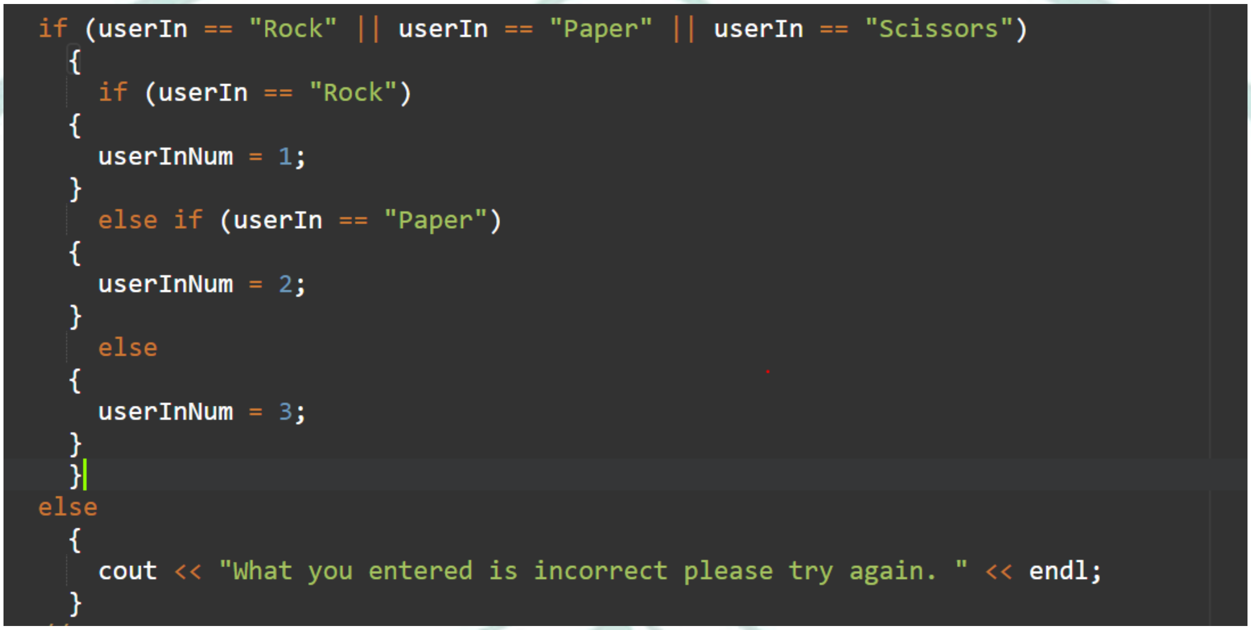
Tip: Remember to use the double equal sign when comparing two things

There should be **three cases**, each one assigning the integer variable that was declared in step 8 to a corresponding value (Rock = 1, Paper = 2, Scissors = 3)

Tip: Remember when setting a variable equal to something use a single equal sign

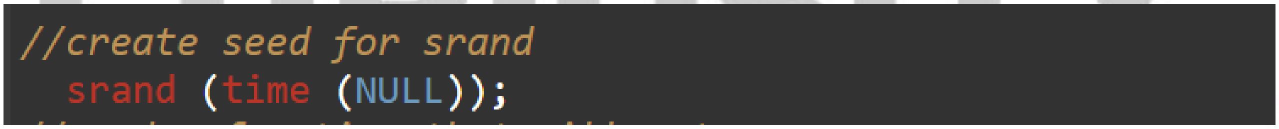
The next portion of the if statement is the else, so if what the user entered does not match “Rock”, “Paper” or “Scissors” then throw an error message.

To see all of step 9, look at the image below.



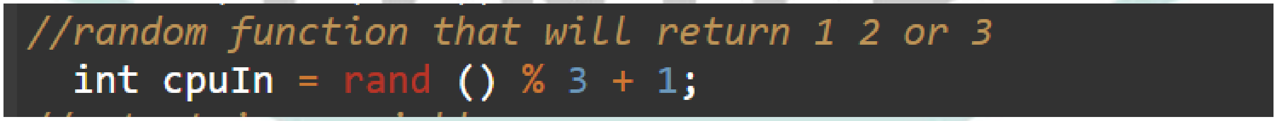
```
if (userIn == "Rock" || userIn == "Paper" || userIn == "Scissors")
  {
    if (userIn == "Rock")
  {
    userInNum = 1;
  }
    else if (userIn == "Paper")
  {
    userInNum = 2;
  }
    else
  {
    userInNum = 3;
  }
  }
else
  {
    cout << "What you entered is incorrect please try again. " << endl;
  }
```

10. **Generate the computers choice** (Rock, Paper or Scissors). First define a seed for the random function to be used in the next step. The Random function will generate a number based off the time pulled from your computer. See the line of code below to generate a seed for rand().



```
//create seed for srand
  srand (time (NULL));
```

11. Now **use the Random function** to return a number stored in an integer variable between 1 and 3.

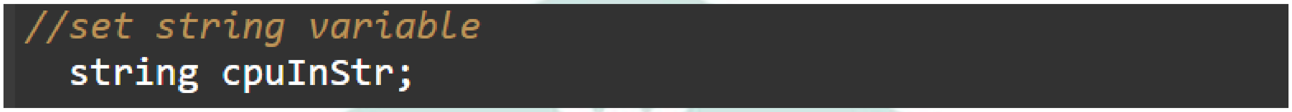


```
//random function that will return 1 2 or 3
  int cpuIn = rand () % 3 + 1;
```

Note: % is modulus, the modulus operator returns the remainder of the random value

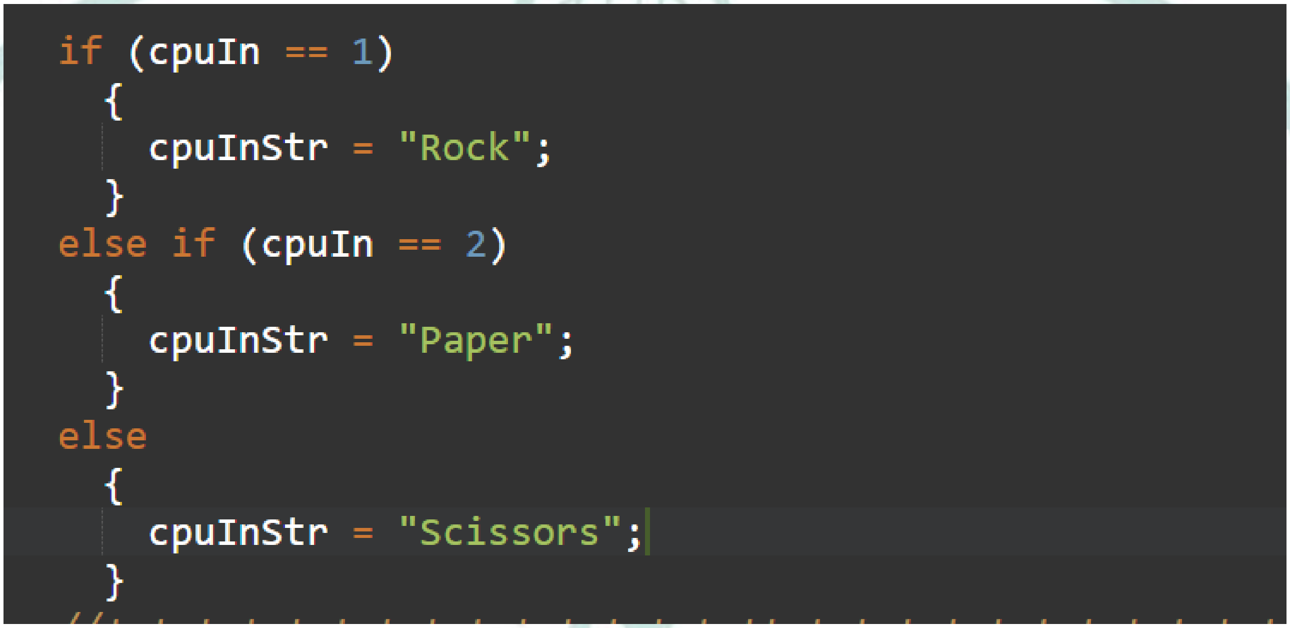
Remember to declare an integer variable to store the number into. In the example above the variable is named cpuIn.

12. **Declare a string variable**. This string variable will be used in another if statement that correlates the computers integer choice to a string (1 = rock, 2 = paper, 3 = scissors)



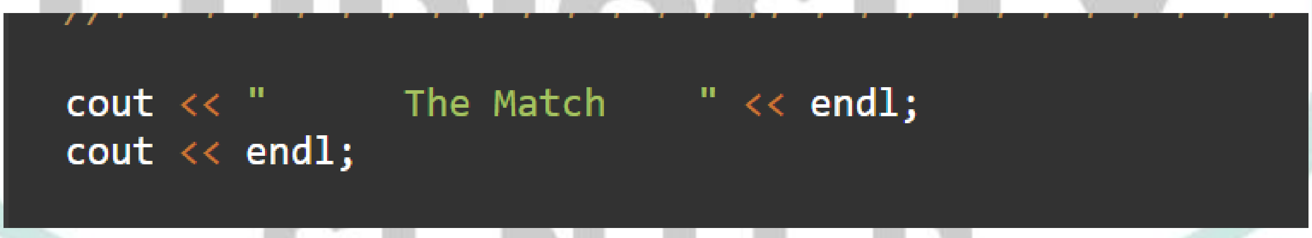
```
//set string variable
  string cpuInStr;
```

13. Set up the **if statement** that checks the integer variable and what the output of the rand function is as a condition, inside set the string variable equal to the corresponding string. See the if statement below.



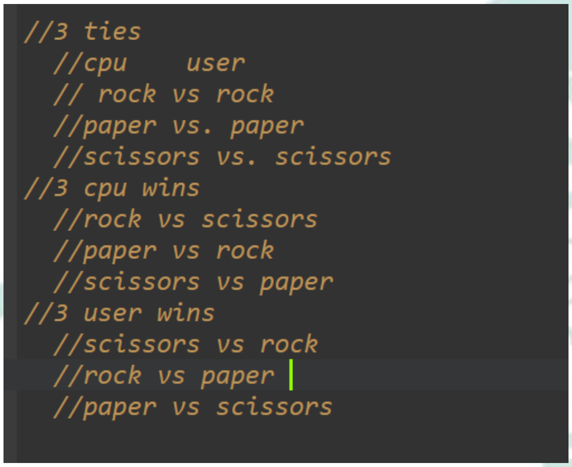
```
if (cpuIn == 1)
  {
    cpuInStr = "Rock";
  }
else if (cpuIn == 2)
  {
    cpuInStr = "Paper";
  }
else
  {
    cpuInStr = "Scissors";
  }
```

14. **Print out** the name of the competition. In this example the programmer chose “The Match”



```
cout << "      The Match    " << endl;
cout << endl;
```

15. **Determine all the different outcomes of the computer going against the user, see below.**



```
//3 ties
  //cpu    user
  // rock vs rock
  //paper vs. paper
  //scissors vs. scissors
//3 cpu wins
  //rock vs scissors
  //paper vs rock
  //scissors vs paper
//3 user wins
  //scissors vs rock
  //rock vs paper
  //paper vs scissors
```

Tip: Remember the basic rules of rock, paper, scissors…

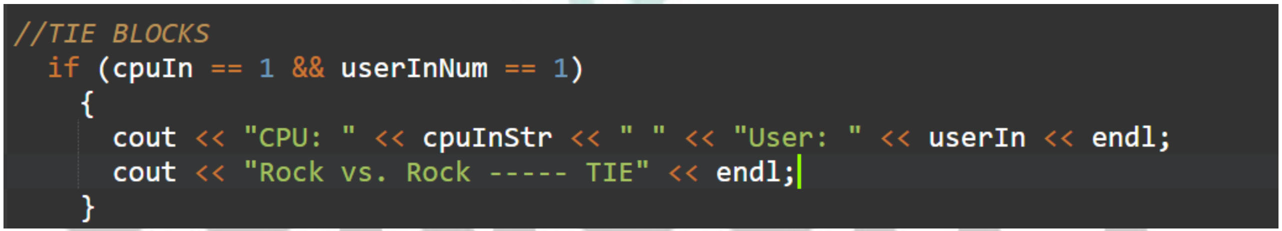
- *Paper covers rock*
- *Rock crushes scissors*
- *Scissors cut paper*

16. Use another if statement to determine which case the computer and the user fall into. For the conditions in these statements use the **and logic operator.**

    Tip: the and operator means that both of the conditions have to be true in order to get into that block of code. The and operator is represented by &&

    Use the cases that were determined above to create blocks of code

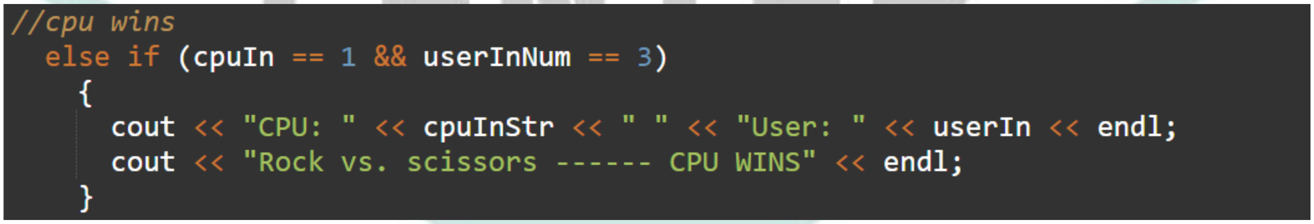
    Tie block example



```
//TIE BLOCKS
  if (cpuIn == 1 && userInNum == 1)
    {
      cout << "CPU: " << cpuInStr << " " << "User: " << userIn << endl;
      cout << "Rock vs. Rock ----- TIE" << endl;
    }
```

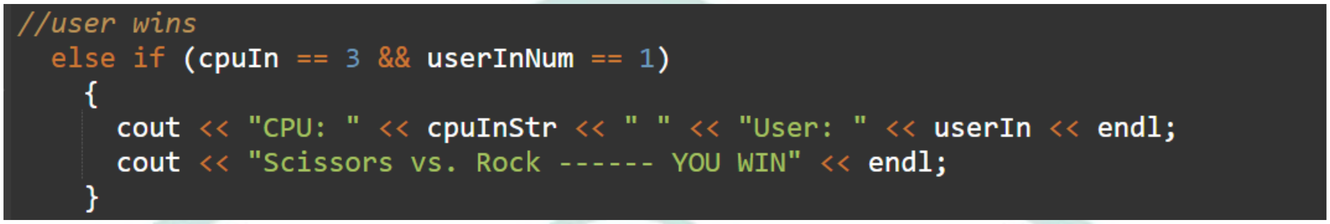
Tip: Concatenation is the technique used to put together all the information in the first cout statement.

CPU wins example



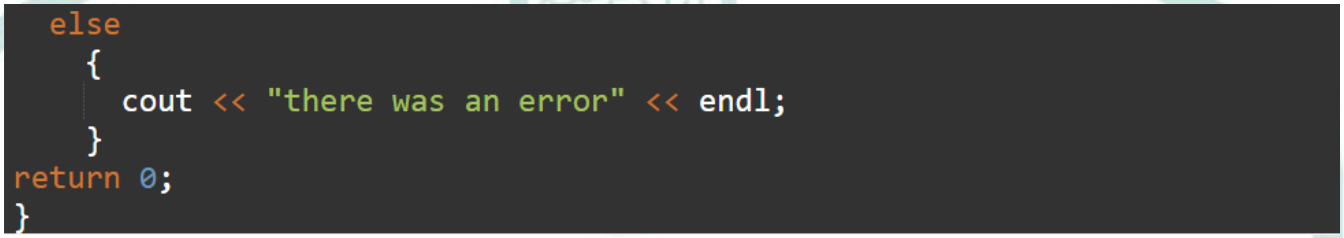
```
//cpu wins
  else if (cpuIn == 1 && userInNum == 3)
    {
      cout << "CPU: " << cpuInStr << " " << "User: " << userIn << endl;
      cout << "Rock vs. scissors ------ CPU WINS" << endl;
    }
```

User wins exapmple



```
//user wins
  else if (cpuIn == 3 && userInNum == 1)
    {
      cout << "CPU: " << cpuInStr << " " << "User: " << userIn << endl;
      cout << "Scissors vs. Rock ------ YOU WIN" << endl;
    }
```

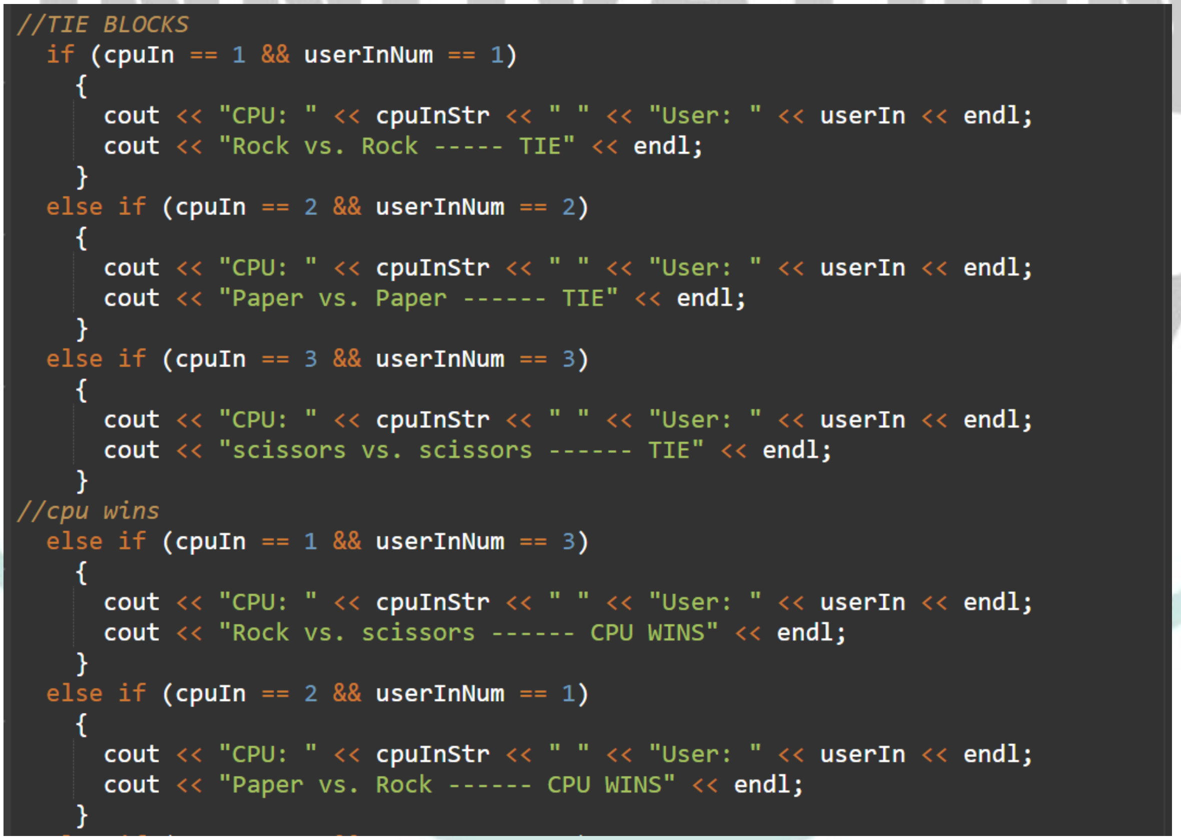
Final case – if there was an error



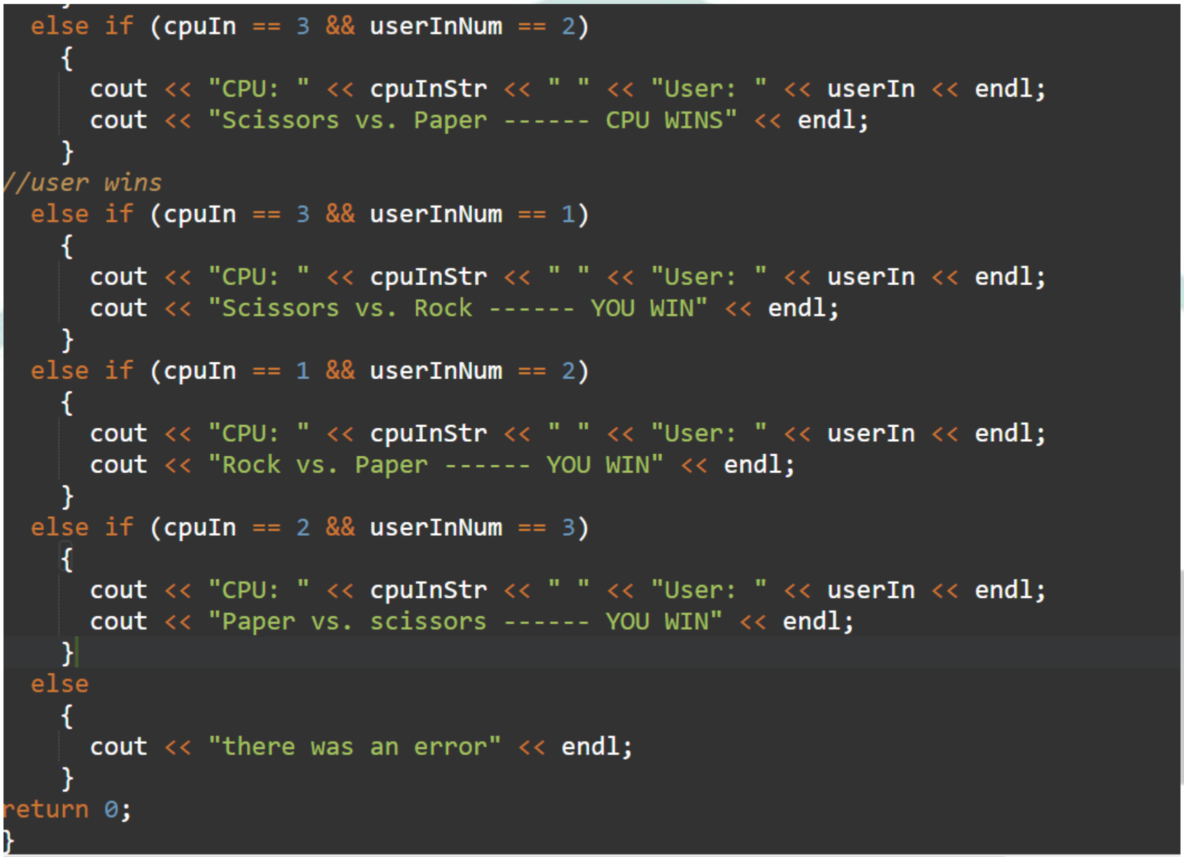
```
  else
    {
      cout << "there was an error" << endl;
    }
return 0;
}
```

Make sure the return statement is on the outside of the last case.

See below for all the cases in step 16:

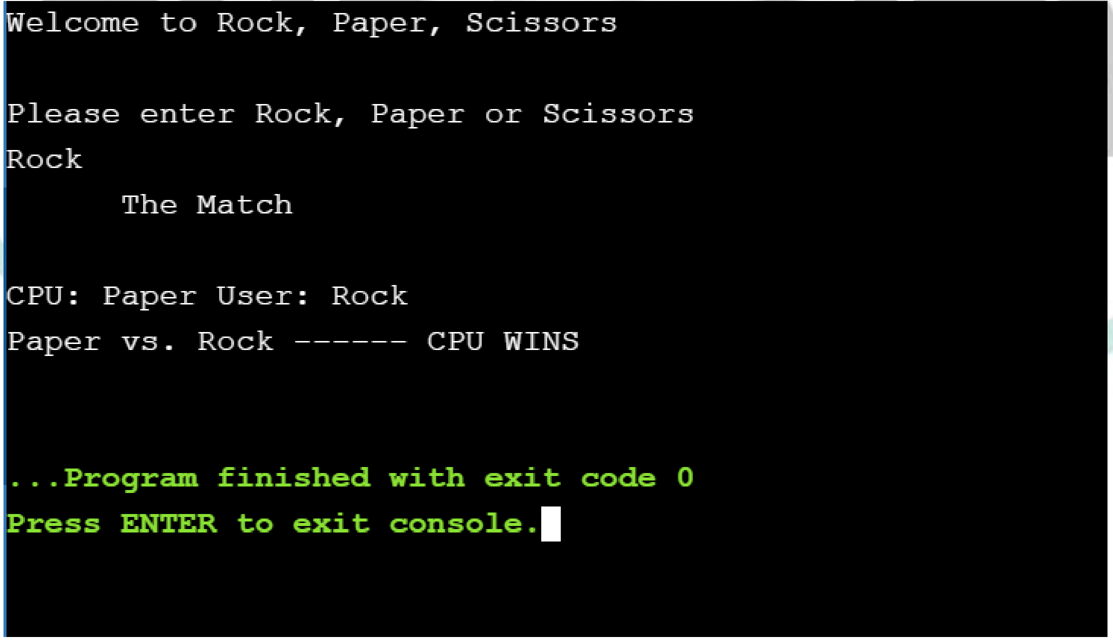


```
//TIE BLOCKS
  if (cpuIn == 1 && userInNum == 1)
    {
      cout << "CPU: " << cpuInStr << " " << "User: " << userIn << endl;
      cout << "Rock vs. Rock ----- TIE" << endl;
    }
  else if (cpuIn == 2 && userInNum == 2)
    {
      cout << "CPU: " << cpuInStr << " " << "User: " << userIn << endl;
      cout << "Paper vs. Paper ------ TIE" << endl;
    }
  else if (cpuIn == 3 && userInNum == 3)
    {
      cout << "CPU: " << cpuInStr << " " << "User: " << userIn << endl;
      cout << "scissors vs. scissors ------ TIE" << endl;
    }
//cpu wins
  else if (cpuIn == 1 && userInNum == 3)
    {
      cout << "CPU: " << cpuInStr << " " << "User: " << userIn << endl;
      cout << "Rock vs. scissors ------ CPU WINS" << endl;
    }
  else if (cpuIn == 2 && userInNum == 1)
    {
      cout << "CPU: " << cpuInStr << " " << "User: " << userIn << endl;
      cout << "Paper vs. Rock ------ CPU WINS" << endl;
    }
```



```
  else if (cpuIn == 3 && userInNum == 2)
    {
      cout << "CPU: " << cpuInStr << " " << "User: " << userIn << endl;
      cout << "Scissors vs. Paper ------ CPU WINS" << endl;
    }
//user wins
  else if (cpuIn == 3 && userInNum == 1)
    {
      cout << "CPU: " << cpuInStr << " " << "User: " << userIn << endl;
      cout << "Scissors vs. Rock ------ YOU WIN" << endl;
    }
  else if (cpuIn == 1 && userInNum == 2)
    {
      cout << "CPU: " << cpuInStr << " " << "User: " << userIn << endl;
      cout << "Rock vs. Paper ------ YOU WIN" << endl;
    }
  else if (cpuIn == 2 && userInNum == 3)
    {
      cout << "CPU: " << cpuInStr << " " << "User: " << userIn << endl;
      cout << "Paper vs. scissors ------ YOU WIN" << endl;
    }
  else
    {
      cout << "there was an error" << endl;
    }
return 0;
}
```

17. Run the program!



```
Welcome to Rock, Paper, Scissors

Please enter Rock, Paper or Scissors
Rock
      The Match

CPU: Paper User: Rock
Paper vs. Rock ------ CPU WINS


...Program finished with exit code 0
Press ENTER to exit console.
```

IF YOUR PROGRAM IS THROWING ERRORS

1. Check all your **semicolons** and **curly braces** throughout the program.
    - **a.** There should be a semi colon after every statement
    - **b.** There should not be a semicolon after your if conditions
    - **c.** Three should not be a semicolon outside of a curly brace
    - **d.** Curly braces need an opening brace { and a closing brace } to enclose all their code
    - **e.** Make sure main has a curly brace right after it, and a closing brace as the last line in the program.
2. Make sure your tokens are going the right way
    - **a.** **<<** is characters/content going out to the user
    - **b.** **>>** is characters/content coming in from the user
3. Make sure to check the **return statement**, the program should be finished with **return 0;**
4. Check to make sure the cout/cin, string, double, endl, if, else if, else are all lowercase.
5. Make sure that your logic statements are correct
    - **a.** **&&** means that both conditions have to be true in order for that block to be satisfied.
    - **b.** **||** means that only one condition have to be true in order for that block to be satisfied.
6. Make sure that **every if statement ends with an else case** that acts as a default if the conditions in the if statement are not met.

FOR MORE INFORMATION

For more information please visit https://c2pipeline.wayne.edu/stem-lab. The full .cpp file of this program can be accessed from here as well!

Happy coding, as always!

-Mikayla