\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* PROGRAMMED BY : BBC & FSC

\* Student ID : 382108

\* Class : CS1A --> Mon/Wed 9:30am

\* LAB #21 :Rock, Paper, Scissors

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter Player's Name: Andrew Daniels

Enter Number of Rounds in Match: 3

R - Rock

P - Paper

S - Scissors

Enter your play: r

Computer chooses Scissors

WINNER WINNER CHICKEN DINNER!

Andrew Daniels wins!

R - Rock

P - Paper

S - Scissors

Enter your play: s

Computer chooses Rock

Better luck next time, Andrew Daniels

R - Rock

P - Paper

S - Scissors

Enter your play: x

\*\* INVALID INPUT - Please Enter (R, P, or S) \*\*R - Rock

P - Paper

S - Scissors

Enter your play: p

Computer chooses Paper

Better luck next time, Andrew Daniels

\*\*\*\*\*\*\*\*\* FINAL RESULTS \*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Andrew Daniels is the winner!!

Andrew Daniels won an average of 33% of the time!

Enter Player's Name: Shawn Azar

Enter Number of Rounds in Match: 5

R - Rock

P - Paper

S - Scissors

Enter your play: r

Computer chooses Scissors

WINNER WINNER CHICKEN DINNER!

Shawn Azar wins!

R - Rock

P - Paper

S - Scissors

Enter your play: p

Computer chooses Scissors

Better luck next time, Shawn Azar

R - Rock

P - Paper

S - Scissors

Enter your play: s

Computer chooses Scissors

Better luck next time, Shawn Azar

R - Rock

P - Paper

S - Scissors

Enter your play: x

\*\* INVALID INPUT - Please Enter (R, P, or S) \*\*R - Rock

P - Paper

S - Scissors

Enter your play: r

Computer chooses Paper

Better luck next time, Shawn Azar

R - Rock

P - Paper

S - Scissors

Enter your play: r

Computer chooses Paper

Better luck next time, Shawn Azar

\*\*\*\*\*\*\*\*\* FINAL RESULTS \*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Shawn Azar is the winner!!

Shawn Azar won an average of 20% of the time!

Enter Player's Name: Erik Karlsson

Enter Number of Rounds in Match: 7

R - Rock

P - Paper

S - Scissors

Enter your play: R

Computer chooses Rock

Better luck next time, Erik Karlsson

R - Rock

P - Paper

S - Scissors

Enter your play: S

Computer chooses Paper

WINNER WINNER CHICKEN DINNER!

Erik Karlsson wins!

R - Rock

P - Paper

S - Scissors

Enter your play: P

Computer chooses Paper

Better luck next time, Erik Karlsson

R - Rock

P - Paper

S - Scissors

Enter your play: x

\*\* INVALID INPUT - Please Enter (R, P, or S) \*\*R - Rock

P - Paper

S - Scissors

Enter your play: R

Computer chooses Scissors

WINNER WINNER CHICKEN DINNER!

Erik Karlsson wins!

R - Rock

P - Paper

S - Scissors

Enter your play: S

Computer chooses Rock

Better luck next time, Erik Karlsson

R - Rock

P - Paper

S - Scissors

Enter your play: P

Computer chooses Rock

WINNER WINNER CHICKEN DINNER!

Erik Karlsson wins!

R - Rock

P - Paper

S - Scissors

Enter your play: k

\*\* INVALID INPUT - Please Enter (R, P, or S) \*\*R - Rock

P - Paper

S - Scissors

Enter your play: s

Computer chooses Scissors

Better luck next time, Erik Karlsson

\*\*\*\*\*\*\*\*\* FINAL RESULTS \*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Erik Karlsson is the winner!!

Erik Karlsson won an average of 43% of the time!

//header file to call upon functions.

**#ifndef** HEADER\_H\_

**#define** HEADER\_H\_

//preprocessor directives for main.cpp and functions.cpp

**#include** <iostream>

**#include** <iomanip>

**#include** <string>

**using** **namespace** std;

//function prototypes

**void** **PrintHeader**(string asName, **char** asType, **int** asNum);

**void** **GetInput**(string& uName, **int**& rounds);

**void** **ValidatePlay**(**char**& move);

**bool** **CheckWin**(**char** move);

**void** **OutputWin**(**bool** won, string uName, **int**& wonTot);

**void** **OutputMatchWinner**(**int** wonTot, **int** rounds, string uName);

**#endif** /\* HEADER\_H\_ \*/

/\*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* PROGRAMMED BY: BBC & FSC

\* CLASS : CS1A

\* SECTION : MW 9:30a

\* Lab #21b : Functions - Rock, Paper, Scissors

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**#include** "Header.h" //calls on the header file to call functions and

// preprocessor directives

**int** **main**()

{

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* FUNCTIONS - Rock, Paper , Scissors

\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\* This program will emulate a game of rock, paper, scissors.

\*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\* INPUT:

\* unName: Name of user

\* rounds: Number of time user wants to play

\*

\* OUTPUT:

\* uName: Name of user

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**char** move; //what move the user plays

string uName; //name of user

**int** rounds; //number of times user wants to play

**bool** won; //bool to determine whether user won or lost to computer

**int** wonTot; //counter adding total amount of wins

**int** count; //count for the for loop

//assign counters to needed values

wonTot = 0;

count = 1;

//Functions-Section that calls upon all the functions

//Creates the the header for programmer

PrintHeader("Rock, Paper, Scissors", 'L', 21);

//Function to get name and rounds from user

GetInput(uName, rounds);

//for loop to repeat game as many times as user asked

**for**(count=1;count<=rounds;++count)

{

//Function that creates menu for user to input move

ValidatePlay(move);

//Function checks whether user or computer won

won=CheckWin(move);

//Function outputs winner of round

OutputWin(won, uName , wonTot);

}

//Function to output total and average amount of wins

OutputMatchWinner(wonTot, rounds, uName);

**return** 0;

}

**#include** "Header.h" //calls on the header file to call functions and

// preprocessor directives

//This function outputs the header

**void** **PrintHeader**(string asName, **char** asType, **int** asNum)

{

cout << left;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout << "\n\* PROGRAMMED BY : BBC & FSC";

cout << "\n\* " << setw(14) << "Student ID" << ": 382108";

cout << "\n\* " << setw(14) << "Class" << ": CS1A --> Mon/Wed 9:30am";

cout << "\n\* ";

**if**(**toupper**(asType) == 'L')

{

cout << "LAB #" << setw(9);

}

**else**

{

cout << "ASSIGNMENT #" << setw(2);

}

cout << asNum << ":" << asName;

cout << "\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n";

cout << right;

}

//This function receives the name and round amount from the user then returns

// nothing

**void** **GetInput**(string& uName, **int**& rounds)

{

//INPUT the information

cout << "Enter Player's Name: ";

**getline**(cin, uName);

cout << "Enter Number of Rounds in Match: ";

cin >> rounds;

cout << **endl** << **endl**;

cin.ignore(100,'\n');

}

//Function to create menu and input the users move and returns his move

**void** **ValidatePlay**(**char**& move)

{

//Inputs the players move

**do**

{

cout << "R - Rock\nP - Paper\nS - Scissors\n";

cout << "Enter your play: ";

cin.get(move);

**if**(move!='R' && move!='P' && move!='S' && move!='r' && move!='p' && move!='s')

cout << "\*\* INVALID INPUT - Please Enter (R, P, or S) \*\*";

cin.ignore(100, '\n');

}**while**(move!='R' && move!='P' && move!='S' && move!='r' && move!='p' && move!='s');

cout << **endl** << **endl**;

}

//Function that determines if the player won or lost then returns the result

// as a bool named won

**bool** **CheckWin**(**char** move)

{

**int** compMove; //The computers move held as a number

**bool** won; //The winner of the round

string compPlay;

won = **false**;

//RNG to determine computers play

**srand**(**time**(NULL));

compMove = (**rand**()%3+1);

//switch statement to determine what move the computer made

**switch**(compMove)

{

**case** 1: compPlay = "Rock";

**break**;

**case** 2: compPlay = "Paper";

**break**;

**case** 3: compPlay = "Scissors";

}

//Outputs what move the computer made

cout << "Computer chooses " << compPlay;

//if statement determining winner

**if**(move=='R' && compMove==3)

won = **true**;

**else** **if**(move=='P' && compMove==1)

won = **true**;

**else** **if**(move=='S' && compMove==2)

won = **true**;

**else** **if**(move=='r' && compMove==3)

won = **true**;

**else** **if**(move=='p' && compMove==1)

won = **true**;

**else** **if**(move=='s' && compMove==2)

won = **true**;

**return** won;

}

//Function that outputs whether user won or lost and returns nothing

**void** **OutputWin**(**bool** won, string uName, **int**& wonTot)

{

//if statement that changes output based on who won

**if** (won)

{

cout << "\n\nWINNER WINNER CHICKEN DINNER!\n" << uName << " wins!"

"\n\n";

++wonTot;

}

**else**

cout << **endl** << **endl** << "Better luck next time, " << uName << **endl** << **endl**;

}

//Function that Calculates the average win rate and outputs the total/average

**void** **OutputMatchWinner**(**int** wonTot, **int** rounds, string uName)

{

**float** average; //average amount of times won

//output the final results after all rounds are played

cout << "\*\*\*\*\*\*\*\*\* FINAL RESULTS \*\*\*\*\*\*\*\*\*\n";

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n\n";

cout << uName << " is the winner!!\n";

average = **float**(wonTot) / rounds \* 100;

cout << uName << " won an average of " << setprecision(0) << fixed

<< average << "% of the time!";

}