**Blake Allard**

```
#include <iostream>
#include <iomanip>
#include<string>
Using namespace std;

String  name;
int     numRounds;
int     roundCount;
Char    userPlay;
Char    pcPlay;
bool    checkWinner;
bool    checkMatchWinner;

int main()
{
        GetInput(name, numRounds);

    for(roundCount = 1  ; roundCount<= numRounds; roundCount++)
    {
userplay = GetAndCheckPlay(&userplay);
  pcPlay GetComputerPlay(&pcPlay);
        checkWinner = CheckWin();
        OutputWin(name);
    }

    Cout << name;
    OutputMatchWinner();
```

FUNCTION PROTOTYPES:

```
void GetInput(string &name, &numRounds);
char GetAndCheckPlay(~~char&userPlay~~);
Char GetComputerPlay(~~char&pcPlay~~);
bool CheckWin(char&userPlay, char&pcPlay);
void OutputWin(string name, bool checkWinner);
void OutputMatchWinner(string name, ~~bool checkMatchWinner~~);
                                    winPercent
```

# FUNCTIONAL DECOMP/PSEUDOCODE

BEGIN

main → INITIALIZE roundCount = 1

function → INPUT name
INPUT numRounds

main — FOR roundCount to numRounds

function — INPUT userplay

function — PROCESSING pcPlay

function — PROCESSING checkWinner

function → OUTPUT name
OUTPUT checkWinner
END FOR

main — OUTPUT name

function → IF checkWinner = true
OUTPUT winPercent
ELSE
OUTPUT lossPercent

END

Jacob Espinosa

```
#include <iostream>
#include <string>
#include <cstdlib.h>
#include <time.h>

string name;
int roundCount;
char userPlay;
int winCount;
int loseCount;
char compPlay;
bool result;
int roundsPlayed;

srand(time(NULL));

void getInput(string &name, int &roundCount);
char GetAndCheckPlay();
char GetComputerPlay();
bool CheckWin(char userPlay, char compPlay);
void OutputWin(string name, bool playResult);
void OutputMatchWinner(string name,
                       double winPercent);
```

Text

```
int main()
{

void GetInput(string &name, int &roundCount);

    for (roundsPlayed=0;
         roundsPlayed <= roundCount
         ++roundsPlayed)
    {

        userPlay = GetAndCheckPlay();
        compPlay = GetComputerPlay();
        if (CheckWin(userPlay, compPlay) = 1
        {
            ++winCount;
            OutputWin(name, true);
        }
        else
        {
            ++loseCount;
            OutputWin(name, false);
        }
    }
    winPercent = winCount(double) / rounds;
    OutputMatchWinner(name, winCount, roundCount);

    return 0;
}
```

COUNT

COUNT

PLAYED

if USERWIN = CHECKWIN

USERWON

USERLOST

Blake Allard & Jacob Espinosa

CS1A

Professor Rousseau

25 November 2024

Comparing Jacob & Blake's E26: Rock, Paper, Scissors

In "Exercise 26", both Jacob and Blake came to the conclusion that Jacob's was a more refined version of the exercise for a few reasons:

- Jacob's int main called "GetAndCheckPlay" & GetComputerPlay" into variables in main
    - Blake's functions were successfully created the same, however did not call the created functions into variables in main.
- Although Jacob's if conditional statement was created, it was created with the incorrect comparisons
    - Blake's condition was not created in main it was only called.
- Jacob's and Blake's "header file" were almost the same in accuracy, Jacob's had a more accurate depiction on which arguments needed to be passed by reference albeit with some incorrect arguments
    - Blake's "header file" was successfully created the same however, some prototypes were unnecessary or were missing pass by reference indications in his arguments
- Blake's functional decomposition/pseudocode depiction of the program was well organized and detailed
    - Jacob's was integrated into his int main and could have been more descriptive

In conclusion, there were pros and cons to each of Blake's and Jacob's exercises varying on the section.  However, overall Jacob showed a more detailed layout and a bit more accuracy in his planning and execution on plotting out his program structure.

```
 1 ************************************************
 2 * PROGRAMMED BY : Blake Allard & Jacob Espinosa
 3 * CLASS         : CS1A
 4 * SECTION       : MW - 8am
 5 * LAB #14       : Rock, Paper, Scissors
 6 ************************************************
 7
 8 Enter Player's Name: Andrew Daniels
 9 Enter Number of Rounds in Match: 3
10
11 **********************
12 ** CHOOSE YOUR PLAY **
13 **********************
14 R - Rock
15 P - Paper
16 S - Scissors
17 Enter your play: R
18
19 Andrew Daniels chooses ROCK!
20 Computer chooses PAPER!
21
22 Computer wins, better luck next time, Andrew Daniels!
23
24
25 **********************
26 ** CHOOSE YOUR PLAY **
27 **********************
28 R - Rock
29 P - Paper
30 S - Scissors
31 Enter your play: s
32
33 Andrew Daniels chooses SCISSOR!
34 Computer chooses PAPER!
35
36 Andrew Daniels WINS!!
37
38
39 **********************
40 ** CHOOSE YOUR PLAY **
41 **********************
42 R - Rock
43 P - Paper
44 S - Scissors
45 Enter your play: x
46
47 ** INVALID INPUT - Please Enter (R, P, or S) **
48
49 R - Rock
50 P - Paper
51 S - Scissors
```

```
Enter your play: P

Andrew Daniels chooses PAPER!
Computer chooses SCISSORS!

Computer wins, better luck next time, Andrew Daniels!


**********************************
********** FINAL RESULTS **********
**********************************

Andrew Daniels lost 67% of the time!


Enter Player's Name: Shawn Azar
Enter Number of Rounds in Match: a

** INVALID INPUT - Please enter a valid number **

Enter Number of Rounds in Match: 5


**********************
** CHOOSE YOUR PLAY **
**********************
R - Rock
P - Paper
S - Scissors
Enter your play: P

Shawn Azar chooses PAPER!
Computer chooses PAPER!

Computer wins, better luck next time, Shawn Azar!


**********************
** CHOOSE YOUR PLAY **
**********************
R - Rock
P - Paper
S - Scissors
Enter your play: r

Shawn Azar chooses ROCK!
Computer chooses PAPER!

Computer wins, better luck next time, Shawn Azar!

```

```
103
104 **********************
105 ** CHOOSE YOUR PLAY **
106 **********************
107 R - Rock
108 P - Paper
109 S - Scissors
110 Enter your play: S
111
112 Shawn Azar chooses SCISSOR!
113 Computer chooses SCISSORS!
114
115 Computer wins, better luck next time, Shawn Azar!
116
117
118 **********************
119 ** CHOOSE YOUR PLAY **
120 **********************
121 R - Rock
122 P - Paper
123 S - Scissors
124 Enter your play: S
125
126 Shawn Azar chooses SCISSOR!
127 Computer chooses SCISSORS!
128
129 Computer wins, better luck next time, Shawn Azar!
130
131
132 **********************
133 ** CHOOSE YOUR PLAY **
134 **********************
135 R - Rock
136 P - Paper
137 S - Scissors
138 Enter your play: R
139
140 Shawn Azar chooses ROCK!
141 Computer chooses PAPER!
142
143 Computer wins, better luck next time, Shawn Azar!
144
145
146 ***********************************
147 ********** FINAL RESULTS **********
148 ***********************************
149
150 Shawn Azar lost 100% of the time!
151
152
153
```

```
Enter Player's Name: Erik Karlsson
Enter Number of Rounds in Match: 7


**********************
** CHOOSE YOUR PLAY **
**********************
R - Rock
P - Paper
S - Scissors
Enter your play: p

Erik Karlsson chooses PAPER!
Computer chooses PAPER!

Computer wins, better luck next time, Erik Karlsson!


**********************
** CHOOSE YOUR PLAY **
**********************
R - Rock
P - Paper
S - Scissors
Enter your play: S

Erik Karlsson chooses SCISSOR!
Computer chooses PAPER!

Erik Karlsson WINS!!


**********************
** CHOOSE YOUR PLAY **
**********************
R - Rock
P - Paper
S - Scissors
Enter your play: s

Erik Karlsson chooses SCISSOR!
Computer chooses SCISSORS!

Computer wins, better luck next time, Erik Karlsson!


**********************
** CHOOSE YOUR PLAY **
**********************
R - Rock
P - Paper
```

```
S - Scissors
Enter your play: r

Erik Karlsson chooses ROCK!
Computer chooses SCISSORS!

Erik Karlsson WINS!!


**********************
** CHOOSE YOUR PLAY **
**********************
R - Rock
P - Paper
S - Scissors
Enter your play: R

Erik Karlsson chooses ROCK!
Computer chooses PAPER!

Computer wins, better luck next time, Erik Karlsson!


**********************
** CHOOSE YOUR PLAY **
**********************
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: S

Erik Karlsson chooses SCISSOR!
Computer chooses SCISSORS!

Computer wins, better luck next time, Erik Karlsson!


**********************
** CHOOSE YOUR PLAY **
**********************
R - Rock
P - Paper
S - Scissors
Enter your play: P
```

```
256
257 Erik Karlsson chooses PAPER!
258 Computer chooses ROCK!
259
260 Erik Karlsson WINS!!
261
262
263 **********************************
264 ********** FINAL RESULTS **********
265 **********************************
266
267 Erik Karlsson lost 57% of the time!
```

```
 1 /*******************************************************************************
 2  * AUTHOR     : Blake Allard , Jacob Espinosa
 3  * STUDENT ID : 358888       , Jacob Espinosa
 4  * LAB #14    : Rock, Paper, Scissors
 5  * CLASS      : CS1A
 6  * SECTION    : MW 8am
 7  * DUE DATE   : 11/25/24
 8  ******************************************************************************/
 9
10 #ifndef HEADER_H_
11 #define HEADER_H_
12
13 #include <iostream>  // Provides cin and cout
14 #include <iomanip>   // Provides setw, setprecision
15 #include <string>    // Provides string operations
16 #include <sstream>   // Provides stringstream functionality
17 #include <time.h>    // Provides time for seeding randomness
18 #include <cstdlib>   // Provides rand()
19 using namespace std;
20
21 /*******************************************************************************
22  * FUNCTION PROTOTYPES
23  ******************************************************************************/
24
25 /*******************************************************************************
26  * OutputClassHeader
27  * -----------------------------------------------------------------------------
28  * Outputs a formatted class heading.
29  * -----------------------------------------------------------------------------
30  * PRE-CONDITIONS
31  *   asName : Assignment name
32  *   asNum  : Assignment number
33  *   asType : Assignment type ('L' for Lab, 'A' for Assignment)
34  * POST-CONDITIONS
35  *   Outputs a class heading to the console.
36  ******************************************************************************/
37
38 string OutputClassHeader(string asName,     // IN - assignment name
39                          int    asNum,      // IN - assignment number
40                          char   asType);    // IN - assignment type
41                                             //      'L' = Lab
42                                             //      'A' = Assignment
43
44 /*******************************************************************************
45  * GetInput
46  * -----------------------------------------------------------------------------
47  * Prompts the user for their name and number of rounds to play.
48  * -----------------------------------------------------------------------------
49  * PRE-CONDITIONS
50  *   INT_ERR_MSG : Error message for invalid integer input.
51  * POST-CONDITIONS
```

```
 52 *   Returns the user name and number of rounds.
 53 ********************************************************************************/
 54
 55 void GetInput(string &name,             // OUT - Player's name
 56               int    &roundNumbers,     // OUT - Number of rounds
 57         const string  INT_ERR_MSG);     // IN  - Error message for invalid input
 58
 59 /*******************************************************************************
 60  * GetAndCheckPlay
 61  * -----------------------------------------------------------------------------
 62  * Prompts the user to select their play (Rock, Paper, or Scissors) and validates
 63  * their input.
 64  * -----------------------------------------------------------------------------
 65  * PRE-CONDITIONS
 66  *   CHAR_ERR_MSG : Error message for invalid character input.
 67  * POST-CONDITIONS
 68  *   Returns the validated character representing the user's play.
 69  ********************************************************************************/
 70
 71 char GetAndCheckPlay(string name,               // IN - Player's Name
 72                const string CHAR_ERR_MSG,      // IN - Error message
 73                const string CHAR_PROMPT);      // IN - Input prompt
 74
 75 /*******************************************************************************
 76  * GetComputerPlay
 77  * -----------------------------------------------------------------------------
 78  * Randomly generates a play for the computer.
 79  * -----------------------------------------------------------------------------
 80  * POST-CONDITIONS
 81  *   Returns the computer's play character ('R', 'P', or 'S').
 82  ********************************************************************************/
 83
 84 char GetComputerPlay();
 85
 86 /*******************************************************************************
 87  * CheckWin
 88  * -----------------------------------------------------------------------------
 89  * Determines if the user has won a round.
 90  * -----------------------------------------------------------------------------
 91  * PRE-CONDITIONS
 92  *   userPlay : The user's play ('R', 'P', or 'S').
 93  *   compPlay : The computer's play ('R', 'P', or 'S').
 94  * POST-CONDITIONS
 95  *   Returns true if the user wins, false otherwise.
 96  ********************************************************************************/
 97
 98 bool CheckWin(char userPlay,         // IN - User's play ('R', 'P', or 'S')
 99               char compPlay);        // IN - Computer's play ('R', 'P', or 'S')
100
101 /*******************************************************************************
102  * OutputWin
```

```
103  * ------------------------------------------------------------------------
104  * Outputs the result of a round, indicating whether the user or computer won.
105  * ------------------------------------------------------------------------
106  * PRE-CONDITIONS
107  *   playResult : True if the user won, false otherwise.
108  * POST-CONDITIONS
109  *   Outputs the result to the console.
110  ************************************************************************/
111
112 void OutputWin(string &name,              // IN - Player's name
113                bool    playResult);       // IN - Result of the play
114
115 /************************************************************************
116  * OutputMatchWinner
117  * ------------------------------------------------------------------------
118  * Outputs the final match results based on win and loss percentages.
119  * ------------------------------------------------------------------------
120  * PRE-CONDITIONS
121  *   winPercent  : The percentage of rounds the user won.
122  *   losePercent : The percentage of rounds the user lost.
123  * POST-CONDITIONS
124  *   Outputs the match results to the console.
125  ************************************************************************/
126
127 void OutputMatchWinner(string &name,           // IN - Player's name
128                        double &winPercent,     // IN - Win percentage
129                        double &losePercent);   // IN - Loss percentage
130
131 /************************************************************************
132  * GetValidatedChar
133  * ------------------------------------------------------------------------
134  * Prompts the user for a character input and validates it.
135  * ------------------------------------------------------------------------
136  * PRE-CONDITIONS
137  *   CHAR_ERR_MSG : Error message for invalid character input.
138  * POST-CONDITIONS
139  *   Returns a validated character.
140  ************************************************************************/
141
142 char GetValidatedChar(string prompt,           // IN & OUT - User input prompt
143                 const string CHAR_ERR_MSG);    // IN       - Error message
144
145 /************************************************************************
146  * GetValidInt
147  * ------------------------------------------------------------------------
148  * Prompts the user for an integer input and validates it.
149  * ------------------------------------------------------------------------
150  * PRE-CONDITIONS
151  *   ERR_MSG : Error message for invalid integer input.
152  * POST-CONDITIONS
153  *   Returns a validated integer.
```

```
154  ******************************************************************************/
155
156 int GetValidInt(string prompt,         // IN - Input prompt
157             const string ERR_MSG);      // IN - Error message for invalid input
158
159 #endif /* HEADER_H_ */
160
```

```
 1 /*******************************************************************************
 2  * AUTHOR     : Blake Allard , Jacob Espinosa
 3  * STUDENT ID : 358888 , 1188930
 4  * LAB #14    : Rock, Paper, Scissors
 5  * CLASS      : CS1A
 6  * SECTION    : MW 8am
 7  * DUE DATE   : 11/25/24
 8  *******************************************************************************/
 9
10 #include "header.h"
11
12 /*******************************************************************************
13  * PROGRAM DESCRIPTION
14  * -----------------------------------------------------------------------------
15  * This program will determine the winner or loser of a game of rock, paper,
16  * scissors, and will output whether the user wins or loses based on the percent
17  * of total wins and loses
18  * -----------------------------------------------------------------------------
19  * INPUT: the user will input their name and the number of rounds they wish to
20  *        play.
21  *
22  * OUTPUT (in loop): based on the user's play, the program will
23  *                   output:
24  *
25  *                  - "Jacob WINS!!"
26  *                  - "Computer wins, better luck next time, Jacob!"
27  *
28  * OUTPUT (out of loop): based on the win count & lose count, the program will
29  *                       output:
30  *
31  *                     - "Jacob lost 33% of the time!"
32  *                     - "Jacob is the WINNER, WINNER, CHICKEN DINNER!!
33  *                        Jacob won 100% of the time!"
34  *
35  *
36  * Example Input/Output:
37  *
38  * Enter Player's Name: Jacob
39  * Enter Number of Rounds in Match: 3
40  *
41  *
42  * **********************
43  * ** CHOOSE YOUR PLAY **
44  * **********************
45  * R - Rock
46  * P - Paper
47  * S - Scissors
48  * Enter your play: R
49  *
50  * Jacob chooses ROCK!
51  * Computer chooses SCISSORS!
```

```
 52 *
 53 * Jacob WINS!!
 54 *
 55 *
 56 * **********************
 57 * ** CHOOSE YOUR PLAY **
 58 * **********************
 59 * R - Rock
 60 * P - Paper
 61 * S - Scissors
 62 * Enter your play: s
 63 *
 64 * Jacob chooses SCISSOR!
 65 * Computer chooses SCISSORS!
 66 *
 67 * Computer wins, better luck next time, Jacob!
 68 *
 69 *
 70 * **********************
 71 * ** CHOOSE YOUR PLAY **
 72 * **********************
 73 * R - Rock
 74 * P - Paper
 75 * S - Scissors
 76 * Enter your play: P
 77 *
 78 * Jacob chooses PAPER!
 79 * Computer chooses PAPER!
 80 *
 81 * Computer wins, better luck next time, Jacob!
 82 *
 83 *
 84 * ***********************************
 85 * ********** FINAL RESULTS **********
 86 * ***********************************
 87 *
 88 * Jacob lost 67% of the time!
 89 ********************************************************************************/
 90
 91 int main()
 92 {
 93
 94    /*****************************************************************************
 95     * CONSTANTS
 96     * ---------------------------------------------------------------------------
 97     * OUTPUT - USED TO ERROR CHECK USERPLAY INPUT
 98     * ---------------------------------------------------------------------------
 99     * ERR_MSG      - OUTPUT ERROR MESSAGE WHEN INVALID CHOICE IS ENTERED
100     * INT_ERR_MSG  - OUTPUT ERROR MESSAGE WHEN AN INVALID NUMBER IS ENTERED
101     * CHAR_ERR_MSG - OUTPUT ERROR MESSAGE WHEN AN INVALID CHARACTER IS ENTERED
102     * CHAR_PROMPT  - PROMPTS THE USER TO SELECT 'R', 'P', OR 'S' AS VALID INPUTS
```

```
103     ******************************************************************************/
104
105     const string ERR_MSG      = "Invalid Input!";      // OUT - invalid input
106
107     const string INT_ERR_MSG  = "\n** INVALID INPUT"
108                                 " - Please enter "
109                                 "a valid number **\n"; // OUT - invalid number
110
111     const string CHAR_ERR_MSG = "\n** INVALID INPUT"
112                                 " - Please Enter "
113                                 "(R, P, or S) **\n";   // OUT - invalid char
114
115     const string CHAR_PROMPT  = "R - Rock\n"
116                                 "P - Paper\n"
117                                 "S - Scissors\n"
118                                 "Enter your play: ";   // OUT - prompt menu
119
120     /******************************************************************************
121      * VARIABLES
122      ******************************************************************************/
123
124     string classHeader;                     //        OUT  - Class description
125     string asName;                          // IN           - Assignment name
126     int    asNum;                           // IN           - Assignment number
127     char   asType;                          // IN           - Assignment type
128     string name;                            // IN   & OUT   - User's name
129     int    roundCount;                      // CALC & OUT   - Number of rounds
130     int    winCount;                        // CALC & OUT   - User rounds won
131     int    loseCount;                       // CALC & OUT   - User rounds lost
132     int    roundNumber;                     // IN   & CALC  - Number of rounds
133     double winPercent;                      // CALC & OUT   - % of rounds won
134     double losePercent;                     // CALC & OUT   - % of rounds lost
135     char   userPlay;                        // CALC & OUT   - User's play choice
136     char   compPlay;                        // CALC & OUT   - Pc's play choice
137     bool   result;                          // CALC         - Result of round
138
139     /******************************************************************************
140      * INITIALIZATIONS
141      ******************************************************************************/
142
143     asName    = "Rock, Paper, Scissors";
144     asNum     =  14;
145     asType    = 'L';
146     winCount  = 0;
147     loseCount = 0;
148
149     /******************************************************************************
150      * OUTPUT - class heading
151      ******************************************************************************/
152
153     classHeader = OutputClassHeader(asName, asNum, asType);
```

```
154
155     cout << classHeader;
156
157     /*****************************************************************************
158      * INPUT - prompt the user for their name and desired # of rounds to play
159      * ---------------------------------------------------------------------------
160      * EXAMPLE:
161      *          Enter Player's Name: Blake
162      *          Enter Number of Rounds in Match: 5
163      *****************************************************************************/
164
165     GetInput(name, roundNumber, INT_ERR_MSG);
166
167     winCount  = 0;
168     loseCount = 0;
169
170     /*****************************************************************************
171      * PROCESSING (IN LOOP) - compares userPlay to compPlay, if the
172      *                        user wins win count is incremented, if the user
173      *                        loses lose count is incremented.
174      *****************************************************************************/
175
176     for (roundCount = 1; roundCount <= roundNumber; roundCount++)
177         {
178             userPlay = GetAndCheckPlay(name, CHAR_ERR_MSG, CHAR_PROMPT);
179             compPlay = GetComputerPlay();
180             result   = CheckWin(userPlay, compPlay);
181
182             if (result)
183             {
184                 ++winCount;
185                 OutputWin(name, result);
186             }
187             else
188             {
189                 ++loseCount;
190                 OutputWin(name, result);
191             }
192         }
193
194     /*****************************************************************************
195      * PROCESSING (OUT OF LOOP) - checks for divide by 0 error, calculates &
196      *                            formats win or lose percent
197      *****************************************************************************/
198
199     if (roundNumber > 0)
200         {
201             //FORMATTING - format decimal place
202             cout << setprecision(0) << fixed;
203             winPercent  = winCount  / (double)roundNumber * 100;
204             losePercent = loseCount / (double)roundNumber * 100;
```

```
205
206         }
207         else
208         {
209             cout << "Exiting Program";
210         }
211
212     /*****************************************************************************
213      * OUTPUT - output the user's name & win percent if they won or lose percent
214      *          if they won
215      *---------------------------------------------------------------------------
216      * EXAMPLE:
217      *         **********************************
218      *         ********** FINAL RESULTS **********
219      *         **********************************
220      *
221      *         Blake is the WINNER, WINNER, CHICKEN DINNER!!
222      *         Blake won 67% of the time!
223      *****************************************************************************/
224
225     OutputMatchWinner(name, winPercent, losePercent);
226
227         //FORMATTING - reset decimal place
228         cout << setprecision(6);
229         cout.unsetf(ios::fixed);
230         return 0;
231     }
232
```

```cpp
/*******************************************************************************
 * AUTHOR     : Blake Allard , Jacob Espinosa
 * STUDENT ID : 358888       , 1188930
 * LAB #14    : Rock, Paper, Scissors
 * CLASS      : CS1A
 * SECTION    : MW 8am
 * DUE DATE   : 11/25/24
 ******************************************************************************/

#include "header.h"

/*******************************************************************************
 * FUNCTION GetInput
 * -----------------------------------------------------------------------------
 * This function prompts the user to enter their name and the number of rounds
 * they want to play. The function validates the number of rounds input to
 * ensure it is a valid integer.
 *
 * -----------------------------------------------------------------------------
 * PRE-CONDITIONS
 *   name          : String to store the player's name.
 *   roundNumber   : Integer variable to store the number of rounds.
 *   INT_ERR_MSG   : Error message for invalid integer input.
 *
 * POST-CONDITIONS
 *   The player's name and the validated number of rounds are returned via
 *   reference parameters.
 ******************************************************************************/

void GetInput(string &name,               // OUT - Player's name
              int    &roundNumber,        // OUT - Number of rounds
              const   string INT_ERR_MSG) // IN  - Error msg for invalid input
{
    cout << "Enter Player's Name: ";
    getline(cin, name);

    cout << "Enter Number of Rounds in Match: ";
    roundNumber = GetValidInt("Enter Number of Rounds in Match: ", INT_ERR_MSG);
    cin.ignore(10000, '\n');
    cout << endl;
}
```

```cpp
/*******************************************************************************
 * AUTHOR     : Blake Allard , Jacob Espinosa
 * STUDENT ID : 358888       , 1188930
 * LAB #14    : Rock, Paper, Scissors
 * CLASS      : CS1A
 * SECTION    : MW 8am
 * DUE DATE   : 11/25/24
 ******************************************************************************/

#include "header.h"

/*******************************************************************************
 * FUNCTION GetAndCheckPlay
 * ----------------------------------------------------------------------------
 * Prompts the user to choose their play (Rock, Paper, or Scissors). This
 * function validates the input and outputs the chosen play in uppercase
 * along with its full name (e.g., "ROCK!").
 *
 * ----------------------------------------------------------------------------
 * PRE-CONDITIONS
 *   name         : The player's name must be passed in.
 *   CHAR_ERR_MSG : The error message to display for invalid input.
 *   CHAR_PROMPT  : The prompt to display for the user's input.
 *
 * POST-CONDITIONS
 *   Returns the validated play character ('R', 'P', or 'S').
 ******************************************************************************/

char GetAndCheckPlay(string name,               // IN - Player's name
                     const string CHAR_ERR_MSG, // IN - Error message
                     const string CHAR_PROMPT)  // IN - Input prompt
{
    char userPlay;                              // OUT - Validated play
    string play;                                // OUT - Full name of the play

    cout << "**********************\n";
    cout << "** CHOOSE YOUR PLAY **\n";
    cout << "**********************\n";
    cout << "R - Rock\n";
    cout << "P - Paper\n";
    cout << "S - Scissors\n";

    cout << "Enter your play: ";
    userPlay = GetValidatedChar(CHAR_ERR_MSG, CHAR_PROMPT);
    cin.ignore(10000, '\n');
    userPlay = toupper(userPlay);

    switch (userPlay)
    {
        case 'R': play = "ROCK!";
              break;
```

```
52          case 'S': play = "SCISSOR!";
53                break;
54          case 'P': play = "PAPER!";
55                break;
56      }
57
58      cout << name << " chooses " << play << endl;
59
60      return userPlay;
61  }
62
63
64
65
66
67
```

```cpp
/*******************************************************************************
 * AUTHOR     : Blake Allard , Jacob Espinosa
 * STUDENT ID : 358888       , 1188930
 * LAB #14    : Rock, Paper, Scissors
 * CLASS      : CS1A
 * SECTION    : MW 8am
 * DUE DATE   : 11/25/24
 *******************************************************************************/

#include "header.h"

/*******************************************************************************
 * FUNCTION GetComputerPlay
 * -----------------------------------------------------------------------------
 * Generates a random play for the computer (Rock, Paper, or Scissors). The
 * function uses the rand() function to pick one of the three plays.
 *
 * -----------------------------------------------------------------------------
 * PRE-CONDITIONS
 *   None
 *
 * POST-CONDITIONS
 *   Returns the randomly chosen play character ('R', 'P', or 'S').
 *******************************************************************************/

char GetComputerPlay()
{
    char   compPlay;           // OUT - Randomly chosen play
    string play;               // OUT - Full name of the play
    int    myRandomValue;      // CALC - Random number for play selection


    myRandomValue = rand() % 3 + 1;

    switch (myRandomValue)
    {
        case 1: compPlay = 'R'; play = "ROCK!"; break;
        case 2: compPlay = 'S'; play = "SCISSORS!"; break;
        case 3: compPlay = 'P'; play = "PAPER!"; break;
    }

    cout << "Computer chooses " << play << "\n\n";

    return compPlay;
}
```

```cpp
/*******************************************************************************
 * AUTHOR     : Blake Allard , Jacob Espinosa
 * STUDENT ID : 358888       , 1188930
 * LAB #14    : Rock, Paper, Scissors
 * CLASS      : CS1A
 * SECTION    : MW 8am
 * DUE DATE   : 11/25/24
 *******************************************************************************/

#include "header.h"

/*******************************************************************************
 * FUNCTION CheckWin
 * -----------------------------------------------------------------------------
 * This function determines if the user has won based on their play and the
 * computer's play. The rules of "Rock, Paper, Scissors" are applied:
 * Rock beats Scissors, Scissors beats Paper, and Paper beats Rock.
 *
 * -----------------------------------------------------------------------------
 * PRE-CONDITIONS
 *   userPlay : The user's choice of 'R', 'P', or 'S' must be passed in.
 *   compPlay : The computer's choice of 'R', 'P', or 'S' must be passed in.
 *
 * POST-CONDITIONS
 *   This function returns true if the user wins, otherwise false.
 *******************************************************************************/

bool CheckWin(char userPlay,   // IN - User's play ('R', 'P', or 'S')
              char compPlay)   // IN - Computer's play ('R', 'P', or 'S')
{
    bool result;              // OUT - Result of the match

    if ((userPlay == 'R' && compPlay == 'S') ||
        (userPlay == 'S' && compPlay == 'P') ||
        (userPlay == 'P' && compPlay == 'R'))
    {
        result = true;
    }
    else
    {
        result = false;
    }

    return result;
}
```

```cpp
/*******************************************************************************
 * AUTHOR     : Blake Allard , Jacob Espinosa
 * STUDENT ID : 358888       , 1188930
 * LAB #14    : Rock, Paper, Scissors
 * CLASS      : CS1A
 * SECTION    : MW 8am
 * DUE DATE   : 11/25/24
 *******************************************************************************/

#include "header.h"

/*******************************************************************************
 * FUNCTION OutputWin
 * -----------------------------------------------------------------------------
 * Outputs the result of a single play, indicating whether the user or the
 * computer won the round.
 *
 * -----------------------------------------------------------------------------
 * PRE-CONDITIONS
 *   name       : Player's name.
 *   playResult : Result of the play (true if user wins, false otherwise).
 *
 * POST-CONDITIONS
 *   Outputs the result of the play.
 *******************************************************************************/

void OutputWin(string &name,        // IN - Player's name
               bool    playResult)  // IN - Result of the play
{
    if (playResult)
    {
        cout << name << " WINS!!\n\n\n";
    }
    else
    {
        cout << "Computer wins, better luck next time, " << name << "!\n\n\n";
    }
}
```

```cpp
/*******************************************************************************
 * AUTHOR     : Blake Allard , Jacob Espinosa
 * STUDENT ID : 358888       , 1188930
 * LAB #14    : Rock, Paper, Scissors
 * CLASS      : CS1A
 * SECTION    : MW 8am
 * DUE DATE   : 11/25/24
 *******************************************************************************/

#include "header.h"

/*******************************************************************************
 * FUNCTION OutputMatchWinner
 * -----------------------------------------------------------------------------
 * Outputs the final winner of the match based on the win and loss percentages.
 *
 * -----------------------------------------------------------------------------
 * PRE-CONDITIONS
 *   name       : Player's name.
 *   winPercent : Percentage of wins.
 *   losePercent: Percentage of losses.
 *
 * POST-CONDITIONS
 *   Outputs the match result message.
 *******************************************************************************/

void OutputMatchWinner(string &name,          // IN - Player's name
                       double &winPercent,    // IN - Win percentage
                       double &losePercent)   // IN - Loss percentage
{
    cout << "***********************************\n";
    cout << "********** FINAL RESULTS **********\n";
    cout << "***********************************\n\n";

    if (winPercent > losePercent)
    {
        cout << name << " is the WINNER, WINNER, CHICKEN DINNER!!\n";
        cout << name << " won " << winPercent << "% of the time!";
    }
    else
    {
        cout << name << " lost " << losePercent << "% of the time!";
    }
}
```

```cpp
/*******************************************************************************
 * AUTHOR     : Blake Allard , Jacob Espinosa
 * STUDENT ID : 358888       , 1188930
 * LAB #14    : Rock, Paper, Scissors
 * CLASS      : CS1A
 * SECTION    : MW 8am
 * DUE DATE   : 11/25/24
 *******************************************************************************/

#include "header.h"

/*****************************************************************
* FUNCTION OutputClassHeader
*----------------------------------------------------------------
* This function receives an assignment name, type
* and number then outputs the appropriate class header -
* returns nothing.
*
*----------------------------------------------------------------
* PRE-CONDITIONS
* asName: Assignment Name
* asNum : Assignment Number
* asType: Assignment Type ==> THIS SHOULD CONTAIN:
* 'L' for Labs
* 'A' for Assignments
*
* POST-CONDITIONS
* This function will output the class heading.
*****************************************************************/

string OutputClassHeader(string asName,        // IN - assignment Name
                         int    asNum,         // IN - assignment number
                         char   asType)        // IN - assignment type
                                               // 'L' = Lab
                                               // 'A' = Assignment
{

    /*************************************************************************
     * CONSTANTS
     * ----------------------------------------------------------------------
     * FORMATTING - used for setw
     * ----------------------------------------------------------------------
     * TITLE_COL : the first column that displays headings for the data
     *************************************************************************/

     const short TITLE_COL = 13;

     /*************************************************************************
     * VARIABLES
     *************************************************************************/

```

```
 52      ostringstream outOss;
 53      string asStr;       // PROC & OUT  - type of assignment (LAB, ASSIGN, etc.)
 54      short  asNumCol;    // CALC & FORM - column width for the assignment number
 55                          //               specific to the type of assignment
 56
 57      /*************************************************************************
 58      * PROCESSING: 1. Assigns the asStr (assignment string) based on the
 59      *                AS_TYPE (assignment type).
 60      *             2. Assigns the asNumCol (assignment column width for the
 61      *                assignment number). The setw will format appropriately
 62      *                based on if this is a lab 'L' or assignment 'A'.
 63      *************************************************************************/
 64
 65      asStr = "ASSIGNMENT";
 66
 67      if (toupper(asType) == 'L')
 68      {
 69          asStr = "LAB";
 70      }
 71
 72      asStr += " #";
 73
 74      asNumCol = TITLE_COL - asStr.length();
 75
 76      /*************************************************************************
 77      * OUTPUT - the class heading table
 78      *
 79      * ********************************************************************
 80      * * PROGRAMMED BY : Blake Allard & Jacob Espinosa
 81      * * CLASS         : CS1A
 82      * * SECTION       : MW - 8am
 83      * * LAB #14       : Rock, Paper, Scissors
 84      * ********************************************************************
 85      *
 86      *************************************************************************/
 87
 88      outOss << left;
 89      outOss << "*************************************************\n";
 90      outOss << "* PROGRAMMED BY : Blake Allard & Jacob Espinosa\n";
 91      outOss << "* "   << setw(TITLE_COL) << "CLASS"   << " : CS1A\n";
 92      outOss << "* "   << setw(TITLE_COL) << "SECTION" << " : MW - 8am\n" ;
 93      outOss << "* "   << asStr << setw(asNumCol) << asNum << " : ";
 94      outOss << asName << endl;
 95      outOss << "*************************************************\n\n";
 96      outOss << right;
 97
 98      return outOss.str();
 99
100 }
101
```