

outputTest.txt

```
1 *****
2 * PROGRAMMED BY : Ali Eshghi
3 * CLASS          : CS1B
4 * SECTION        : MW: 7:30p - 9:50p
5 * LAB #1         : Functions and Arrays
6 *****
7
8 What input file would you like to use? inFile.txt
9 What output file would you like to use? outFile.txt
10
11 MENU OPTIONS
12
13 1 - Find the larger balance
14 2 - Find the smaller balance
15 3 - Obtain sum of all balances
16 4 - Obtain the average of all balances
17 5 - Find Person
18 0 - Exit
19 Enter an option (0 to exit) : 1
20
21 Finding the larger balance...
22
23
24 MENU OPTIONS
25
26 1 - Find the larger balance
27 2 - Find the smaller balance
28 3 - Obtain sum of all balances
29 4 - Obtain the average of all balances
30 5 - Find Person
31 0 - Exit
32 Enter an option (0 to exit) : 2
33 Finding the smaller balance...
34
35
36 MENU OPTIONS
37
38 1 - Find the larger balance
39 2 - Find the smaller balance
40 3 - Obtain sum of all balances
41 4 - Obtain the average of all balances
42 5 - Find Person
43 0 - Exit
44 Enter an option (0 to exit) : 3
45 Obtaining the sum of all balances...
46
47
48 MENU OPTIONS
49
50 1 - Find the larger balance
51 2 - Find the smaller balance
52 3 - Obtain sum of all balances
53 4 - Obtain the average of all balances
54 5 - Find Person
55 0 - Exit
```

outputTest.txt

```
56 Enter an option (0 to exit) : 4
57
58 Obtaining the average of all balances...
59
60
61 MENU OPTIONS
62
63 1 - Find the larger balance
64 2 - Find the smaller balance
65 3 - Obtain sum of all balances
66 4 - Obtain the average of all balances
67 5 - Find Person
68 0 - Exit
69 Enter an option (0 to exit) : 5
70
71 Who do you want to search for (enter done to exit) : Steve Woolston
72 Found.
73
74
75 MENU OPTIONS
76
77 1 - Find the larger balance
78 2 - Find the smaller balance
79 3 - Obtain sum of all balances
80 4 - Obtain the average of all balances
81 5 - Find Person
82 0 - Exit
83 Enter an option (0 to exit) : 5
84
85 Who do you want to search for (enter done to exit) : Jacques Rousseau
86 Jacques Rousseau was not found.
87
88
89 MENU OPTIONS
90
91 1 - Find the larger balance
92 2 - Find the smaller balance
93 3 - Obtain sum of all balances
94 4 - Obtain the average of all balances
95 5 - Find Person
96 0 - Exit
97 Enter an option (0 to exit) : 5
98
99 Who do you want to search for (enter done to exit) : Chris Carroll
100 Found.
101
102
103 MENU OPTIONS
104
105 1 - Find the larger balance
106 2 - Find the smaller balance
107 3 - Obtain sum of all balances
108 4 - Obtain the average of all balances
109 5 - Find Person
110 0 - Exit
```

outputTest.txt

```
111 Enter an option (0 to exit) : 5
112
113 Who do you want to search for (enter done to exit) : Pete McBride
114 Found.
115
116
117 MENU OPTIONS
118
119 1 - Find the larger balance
120 2 - Find the smaller balance
121 3 - Obtain sum of all balances
122 4 - Obtain the average of all balances
123 5 - Find Person
124 0 - Exit
125 Enter an option (0 to exit) : 5
126
127 Who do you want to search for (enter done to exit) : Jean Rousseau
128 Found.
129
130
131 MENU OPTIONS
132
133 1 - Find the larger balance
134 2 - Find the smaller balance
135 3 - Obtain sum of all balances
136 4 - Obtain the average of all balances
137 5 - Find Person
138 0 - Exit
139 Enter an option (0 to exit) : 5
140
141 Who do you want to search for (enter done to exit) : Florence Cyr
142 Florence Cyr was not found.
143
144
145 MENU OPTIONS
146
147 1 - Find the larger balance
148 2 - Find the smaller balance
149 3 - Obtain sum of all balances
150 4 - Obtain the average of all balances
151 5 - Find Person
152 0 - Exit
153 Enter an option (0 to exit) : 0
154
155
156 Thank you for using my program.
```

outFile.txt

1 Larger Balance:

2 ID #	NAME	BALANCE DUE
3 ----	-----	-----
4 1002	Steve Woolston	\$ 1423.2

5

6 Smaller Balance:

7 ID #	NAME	BALANCE DUE
8 ----	-----	-----
9 1001	Jean Rousseau	\$ 15.5

10

11 Sum of Balances for all persons :

12 \$ 4080.48

13

14 Average Balance for all persons

15 \$ 408.05

16

17 Search Name :

18 ID #	NAME	BALANCE DUE
19 ----	-----	-----
20 1002	Steve Woolston	\$ 1423.20

21

22 Search Name :

23

24 Name Not Found in the list.

25

26 Search Name :

27 ID #	NAME	BALANCE DUE
28 ----	-----	-----
29 1008	Chris Carroll	\$ 32.35

30

31 Search Name :

32 ID #	NAME	BALANCE DUE
33 ----	-----	-----
34 1007	Pete McBride	\$ 500.32

35

36 Search Name :

37 ID #	NAME	BALANCE DUE
38 ----	-----	-----
39 1001	Jean Rousseau	\$ 15.50

40

41 Search Name :

42

43 Name Not Found in the list.

44

45

MyHeader.h

```
1 /*****
2  * PROGRAMMER : Ali Eshghi
3  * STUDENT ID : 1112261
4  * CLASS      : CS1B
5  * SECTION    : MW 7:30pm
6  * Assign #1  : Functions and arrays
7  * DUE DATE   : 19 September 2019
8  *****/
9 #ifndef MYHEADER_H_
10 #define MYHEADER_H_
11
12 #include <iostream> // input output
13 #include <iomanip>   // Calculations
14 #include <stdio.h>   //
15 #include <fstream>   // input and output files
16 #include <string>    // strings
17 #include <sstream>   // ostringstream
18 using namespace std;
19
20
21 /*****
22  * Function : PrintHeader
23  * -----
24  * This function stores the header file into a string variable and returns that
25  * string variable to a string variable in the main called "header"
26  *****/
27
28 string PrintHeader(const string PROGRAMMER,
29                   const string CLASS,
30                   const string SECTION,
31                   const int    ASSIGN_NUM,
32                   const string ASSIGN_NAME);
33
34
35 /*****
36  * Function : Input
37  * -----
38  * This function gets the data from the input file and puts them in order in
39  * the parallel arrays for the name, id, and balance using a while loop. This
40  * is a void type function, so it doesn't need any return.
41  *****/
42
43 void Input(ifstream &inFile,
44           const int AR_SIZE,
45           string name[],
46           int id[],
47           float balance[]);
48
49
50 /*****
51  * Function : SearchBalance
52  * -----
53  * This function searches for the largest or smaller balance in the balance
54  * array using a for loop and then returns an integer type variable as the
55  * index of the larger balance in the array to an integer type variable in
```

MyHeader.h

```

56 * the main named "balanceIndex".
57 *****/
58
59 int SearchBalance(float balanceAr[],
60                 int option,
61                 const int AR_SIZE);
62
63
64 /*****
65 * Function : SumOrAvg
66 * -----
67 * This function gets all the balances from the list using a for loop
68 * and add them up together to get the sum and average of all the balances.
69 * then returns a float type variable to a float variable in the main called
70 * "sumOrAvg".
71 *****/
72
73 float SumOrAvg(float balanceAr[],
74               int option,
75               const int AR_SIZE);
76
77
78 /*****
79 * Function : SearchName
80 * -----
81 * This function prompts the user that who is the user searching for and uses
82 * while loop and an boolean expression to check the name array and if the name
83 * was found, returns the index of the array to be used in the parallel arrays
84 * and if the name was not found, lets the user know
85 *****/
86
87 int SearchName(string nameAr[],
88               const int AR_SIZE);
89
90
91 /*****
92 * Function : FileOutput
93 * -----
94 * This function will out put the results that were assigned to variables in
95 * the main from the other functions to the output file. the function is a void
96 * type function, so it doesn't need any return.
97 *****/
98
99 void FileOutput(ostream &outFile,
100               string nameAr[],
101               int idAr[],
102               float balanceAr[],
103               int option,
104               int balanceIndex,
105               float sumOrAvg,
106               int nameIndex);
107
108 #endif /* MYHEADER_H_ */
109

```

main.cpp

```
1 /*****
2  * PROGRAMMER : Ali Eshghi
3  * STUDENT ID : 1112261
4  * CLASS      : CS1B
5  * SECTION    : MW 7:30pm
6  * Assign #1  : Functions and arrays
7  * DUE DATE   : 19 September 2019
8  *****/
9 #include "MyHeader.h"
10
11
12 int main()
13 {
14     /*****
15     * CONSTANTS
16     * -----
17     * OUTPUT - USED FOR CLASS HEADING
18     * -----
19     * PROGRAMMER : Programmer's Name
20     * CLASS      : Student's Course
21     * SECTION    : Class Days and Time
22     * LAB_NUM    : Lab Number (specific to this lab)
23     * LAB_NAME   : Title of the Assignment
24     *****/
25     const string PROGRAMMER = "Ali Eshghi";
26     const string CLASS      = "CS1B";
27     const string SECTION= "MW: 7:30p - 9:50p";
28     const int  ASSIGN_NUM   = 1;
29     const string ASSIGN_NAME= "Functions and Arrays";
30
31     /*****
32     * VARIABLES *
33     *****/
34
35     const int AR_SIZE = 10; //PROCESS - used for size of arrays.
36
37     ofstream outFile ; //OUTPUT - used to save the output.
38     ifstream inFile; //INPUT - gives the data to program.
39     string  inFileName; //INPUT - user input for the infile name.
40     string  outFileName; //INPUT - user input for the outfile name.
41     string  header; //OUTPUT - saves the header file and outputs it.
42     int  option; //IN AND PROCESS - user input for the menu option.
43     int  balanceIndex; //PROCESS - used for the balance search.
44     float  sumOrAvg; //PROCESS - used for getting the sum or average.
45     int  nameIndex; //PROCESS - used for searching the name.
46
47     string nameAr[AR_SIZE]; //PROCESS - stores the names from the input file.
48     int  idAr[AR_SIZE]; //PROCESS - stores the id's from the input file.
49     float  balanceAr[AR_SIZE]; //PROCESS - stores the balances from the input file.
50
51     //the string type variable named "header" saves the header
52     //file as a string inside itself, then outputs that.
53     header = PrintHeader(PROGRAMMER, CLASS, SECTION, ASSIGN_NUM, ASSIGN_NAME);
54     cout << header;
55     outFile << header;
```

```

56
57 //the program prompts the user for the input file and output file names
58 //and then save them as a string and use that to open the file with the
59 //name that the user inputs.
60 cout << left;
61 cout << setw(40) << "What input file would you like to use?";
62 getline(cin,inFileName);
63 inFile.open(inFileName.c_str());
64
65 cout << setw(40) << "What output file would you like to use?";
66 getline(cin,outFileName);
67 outFile.open(outFileName.c_str());
68
69 //The "Input" function gets the data from the input file named by
70 //the user and puts them in three parallel arrays to be used in
71 //the program.
72 Input(inFile, AR_SIZE, nameAr, idAr, balanceAr);
73
74
75 /*****
76 * PROCESS & OUTPUT - Open output file and outputs the data based on the
77 * options that the user inputed in the program.
78 *****/
79
80 //The program shows the menu option to the user and prompts the
81 //user to which option the user would like to use.
82 cout << endl;
83 cout << "MENU OPTIONS" << endl << endl;
84 cout << "1 - Find the larger balance" << endl;
85 cout << "2 - Find the smaller balance" << endl;
86 cout << "3 - Obtain sum of all balances" << endl;
87 cout << "4 - Obtain the average of all balances" << endl;
88 cout << "5 - Find Person" << endl;
89 cout << "0 - Exit" << endl;
90
91 cout << "Enter an option (0 to exit) : ";
92 cin >> option;
93
94 /*****
95 * PROCESS - Begins to use if and else statement based on the user input for
96 * the "option" and chooses the right function to do the work.
97 *****/
98
99 while (option != 0)//While loop that continues to run the program
100 {
101     //until the user inputs "0" as an option.
102     if (option == 1 || option == 2)//If statement for the first two
103     {
104         //options, because they are in same
105         //functions.
106         //This function gets the index of the larger balance
107         //or the smaller balance from the array and returns it
108         // to the variable named "balanceIndex".
109         balanceIndex = SearchBalance(balanceAr, option, AR_SIZE);
110

```


main.cpp

```

111     }//End of the if statement for the first two options
112
113     else if (option == 3 || option == 4)//If statement for the second
114     {                                     //two options because they are
115                                         //in the same function.
116
117         //This function gets the sum or the average of all balances
118         // and returns the value to the variable named "sumOrAvg".
119         sumOrAvg = SumOrAvg(balanceAr, option, AR_SIZE);
120
121     }//End of if statement for the second two options
122
123     else if (option == 5)//If statement for the last option
124     {
125         //This function ask the user for who is the user is searching for
126         //from the list, then searches for the name in the list and if the
127         //name was found, it passes the name's index from the array to the
128         //variable named "nameIndex" and prints out "found", if the name
129         //doesn't exists in the list, then it prints out the name and
130         //"was not found".
131         nameIndex = SearchName(nameAr, AR_SIZE);
132
133     }//End of if statement for the last option
134
135     /*****
136     * OUTPUT – This function outputs the result in the output file.
137     *****/
138
139     //This function outputs all the data acquired from the arrays by
140     //the user from the menu to an output file.
141     FileOutput(outFile, nameAr, idAr, balanceAr, option, balanceIndex,
142     sumOrAvg, nameIndex);
143
144     //The program shows the menu again and prompts the user
145     //to change the option for the while loop.
146     cout << endl;
147     cout << "MENU OPTIONS" << endl;
148     cout << "1 – Find the larger balance" << endl;
149     cout << "2 – Find the smaller balance" << endl;
150     cout << "3 – Obtain sum of all balances" << endl;
151     cout << "4 – Obtain the average of all balances" << endl;
152     cout << "5 – Find Person" << endl;
153     cout << "0 – Exit" << endl;
154
155     cout << "Enter an option (0 to exit) : ";
156     cin >> option;
157 }
158
159 //This message is shown to show the user that the program is done
160 //completely and it is terminated.
161 cout << "\n\nThank you for using my program.";
162 cout << right;
163
164 //These are for closing the files because the program has ended

```

main.cpp

```
165     inFile.close();
166     outFile.close();
167
168     //Returning "0" because of the int main.
169     return 0;
170
171 }
172
```

Input.cpp

```
1 /*****
2  * PROGRAMMER : Ali Eshghi
3  * STUDENT ID : 1112261
4  * CLASS      : CS1B
5  * SECTION    : MW 7:30pm
6  * Assign #1  : Functions and arrays
7  * DUE DATE   : 19 September 2019
8  *****/
9 #include "MyHeader.h"
10
11 /*****
12  * Function : Input
13  * -----
14  * This function gets the data from the input file and puts them in order in
15  * the parallel arrays for the name, id, and balance using a while loop. This
16  * is a void type function, so it doesn't need any return.
17  *****/
18
19
20 void Input(ifstream &iFile,
21           const int AR_SIZE,
22           string nameAr[],
23           int idAr[],
24           float balanceAr[])
25 {
26     /*****
27      * VARIABLES *
28      *****/
29
30     int index; //PROCESS - Used for initializing the arrays
31
32     //Initializing the index to 0 for the while loop
33     index = 0;
34
35     //This while loop uses the input file and the arrays to
36     //get the data from the input file and puts that in the
37     //parallel arrays for the names, id's, and balances.
38     //This while loop doesn't exit before the index reaches
39     //the size of the array and the input file still has data.
40     while(iFile && index < AR_SIZE)
41     {
42         getline(iFile, nameAr[index]);
43         iFile >> idAr[index];
44         iFile >> balanceAr[index];
45         iFile.ignore(100, '\n');
46
47         //increments the index to change the variable in the
48         //while loop for check.
49         index++;
50     }
51 }
52
53
54
55 }
```


FileOutput.cpp

```

1 /*****
2  * PROGRAMMER : Ali Eshghi
3  * STUDENT ID : 1112261
4  * CLASS      : CS1B
5  * SECTION    : MW 7:30pm
6  * Assign #1  : Functions and arrays
7  * DUE DATE   : 19 September 2019
8  *****/
9 #include "MyHeader.h"
10
11 /*****
12  * Function : FileOutput
13  * -----
14  * This function will out put the results that were assigned to variables in
15  * the main from the other functions to the output file. the function is a void
16  * type function, so it doesn't need any return.
17  *****/
18
19 void FileOutput(ostream &outFile,
20                 string  nameAr[],
21                 int     idAr[],
22                 float   balanceAr[],
23                 int     option,
24                 int     balanceIndex,
25                 float   sumOrAvg,
26                 int     nameIndex)
27 {
28
29     //This if statement is for when the user chooses the option 1. It outputs
30     //the larger balance along side of the name and id of the person who has
31     //the larger balance. It uses the integer returned from the "SearchBalance"
32     //function to use in the parallel arrays to get the right data.
33     if (option == 1)
34     {
35         cout    << "\nFinding the larger balance..." << endl << endl;
36         outFile << left;
37         outFile << "Larger Balance:\n";
38         outFile << setw(9) << "ID #" << setw(25) << "NAME" << "BALANCE DUE\n";
39         outFile << setw(9) << "----" << setw(25) << "-----" <<
40         "-----" << endl;
41         outFile << setw(9) << idAr[balanceIndex] << setw(25) <<
42         nameAr[balanceIndex]
43         << "$" << right << setw(10) << balanceAr[balanceIndex] <<
44         endl << endl;
45         outFile << right;
46     } //End of if statement
47
48     //This if statement is for when the user chooses the option 2. It outputs
49     //the smaller balance along side of the name and id of the person who has
50     //the smaller balance. It uses the integer returned from the "SearchBalance"
51     //function to use in the parallel arrays to get the right data.
52     else if (option == 2)
53     {
54         cout    << "Finding the smaller balance..." << endl << endl;

```

FileOutput.cpp

```

53     outFile << left;
54     outFile << "Smaller Balance:\n";
55     outFile << setw(9) << "ID #" << setw(25) << "NAME" << "BALANCE DUE\n";
56     outFile << setw(9) << "----" << setw(25) << "-----" <<
"-----" << endl;
57     outFile << setw(9) << idAr[balanceIndex] << setw(25) <<
nameAr[balanceIndex]
58     << "$" << right << setw(10) << balanceAr[balanceIndex] <<
endl << endl;
59     outFile << right;
60 }//End of if statement
61
62 //This if statement is for when the user chooses the option 3. It outputs
63 //the sum of all balances using the float number returned to main from the
64 // function named "SumOrAvg".
65 else if (option == 3)
66 {
67     cout << "Obtaining the sum of all balances..." << endl << endl;
68     outFile << left;
69     outFile << "Sum of Balances for all persons :\n";
70     outFile << setw(10) << "$" << sumOrAvg << endl << endl;
71     outFile << right;
72 }//End of if statement
73
74 //This if statement is for when the user chooses the option 4. It outputs
75 //the average of all balances using the float number returned to main from
76 //the function named "SumOrAvg".
77 else if (option == 4)
78 {
79     cout << "Obtaining the average of all balances..." << endl << endl;
80     outFile << left;
81     outFile << "Average Balance for all persons\n";
82     outFile << setprecision(2) << fixed;
83     outFile << setw(10) << "$" << sumOrAvg << endl << endl;
84     outFile << right;
85 }//End of if statement
86
87 //This if statement is for when the user chooses the option 5. It outputs
88 //the name and id and balance of the person searched by the user in the
89 //function called "SearchName", using the integer returned from the function
90 //as the integer in the parallel arrays.
91 else if (option == 5)
92 {
93     outFile << left;
94     outFile << "Search Name :" << endl;
95
96     //This if statement indicated that if the integer returned from the
97     //"SearchName" function is between 0 - 9, then it can output the name,
98     //id, and balance of the person based on the parallel arrays.
99     if (nameIndex >= 0 && nameIndex <10)
100     {
101
102         outFile << setw(9) << "ID #" << setw(25) << "NAME" << "BALANCE
DUE\n";
103         outFile << setw(9) << "----" << setw(25) << "-----"

```

FileOutput.cpp

```
104         << "-----" << endl;
105         outFile << setw(9) << idAr[nameIndex] << setw(25) << nameAr[nameIndex]
106         << "$" << right << setw(10) << balanceAr[nameIndex] <<
    endl << endl;
107         outFile << right;
108     }//End of if statement
109
110     //else statement indicates that if the integer returned from the
111     //"SearchName" function is bigger than 9, that means there are
112     //no names in the list matching the searched name.
113     else
114     {
115         outFile << setw(9) << endl;
116         outFile << "Name Not Found in the list." << endl << endl;
117
118     }//End of else statement
119
120     outFile << right;
121 }
122
123
124
125 }
126
```

SearchBalance.cpp

```
1 /*****
2  * PROGRAMMER : Ali Eshghi
3  * STUDENT ID : 1112261
4  * CLASS      : CS1B
5  * SECTION    : MW 7:30pm
6  * Assign #1  : Functions and arrays
7  * DUE DATE   : 19 September 2019
8  *****/
9 #include "MyHeader.h"
10
11 /*****
12  * Function : SearchBalance
13  * -----
14  * This function searches for the largest or smaller balance in the balance
15  * array using a for loop and then returns an integer type variable as the
16  * index of the larger balance in the array to an integer type variable in
17  * the main named "balanceIndex".
18  *****/
19
20 int SearchBalance(float balanceAr[],
21                  int option,
22                  const int AR_SIZE)
23 {
24     /*****
25     * VARIABLES *
26     *****/
27
28     int balanceIndex; // PROCESS - Adjusts to the index of the larger balance
29     int index;        // PROCESS - Used in the for loop for the initial, check
30                       // and change in the loop.
31
32     //Initializing this to 0, to be changed later.
33     balanceIndex = 0;
34
35     //If statement used if the user chooses the option 1.
36     if(option == 1)
37     {
38         //For loop to search the largest balance between the
39         //elements of the array, checking index by index.
40         for (index = 0; index < AR_SIZE; index++)
41         {
42             //If statement to change the integer as the the
43             //index for the larger balance if the balance of the
44             //next element is higher.
45             if (balanceAr[index] > balanceAr[balanceIndex])
46             {
47                 balanceIndex = index;
48             } //End of if statement.
49
50         } //End of the for loop.
51
52     } //End of the if statement.
53
54     //If statement used if the user chooses the option 2.
```


SearchBalance.cpp

```
56  else if(option == 2)
57  {
58      //For loop to search the smaller balance between the
59      //elements of the array, checking index by index.
60      for (index = 1; index <= AR_SIZE; index++)
61      {
62          //If statement to change the integer as the the
63          //index for the larger balance if the balance of the
64          //next element is higher.
65          if (balanceAr[index] < balanceAr[balanceIndex])
66          {
67              balanceIndex = index;
68          }//End of if statement.
69
70
71      }//End of for loop.
72
73  }//End of if statement.
74
75
76
77  //Returns the index integer to the main.
78  return balanceIndex;
79 }
80
81
82
```

SumOrAvg.cpp

```
1 /*****
2  * PROGRAMMER : Ali Eshghi
3  * STUDENT ID : 1112261
4  * CLASS      : CS1B
5  * SECTION    : MW 7:30pm
6  * Assign #1   : Functions and arrays
7  * DUE DATE   : 19 September 2019
8  *****/
9 #include "MyHeader.h"
10
11 /*****
12  * Function : SumOrAvg
13  * -----
14  * This function gets all the balances from the list using a for loop
15  * and add them up together to get the sum and average of all the balances.
16  * then returns a float type variable to a float variable in the main called
17  * "sumOrAvg".
18  *****/
19
20
21 float SumOrAvg(float balanceAr[], int option, const int AR_SIZE)
22 {
23     /*****
24      * VARIABLES *
25      *****/
26
27     float sum; //PROCESS - sum of all numbers
28     float avg; //PROCESS - average of all numbers
29     float sumOrAvg; //PROCESS & OUT - assigns sum or average to itself
30     int index; //PROCESS - used for the for loop.
31
32     //initialize the sum to zero, then adds the balance to it
33     sum = 0;
34
35     //initialize the sum or average to zero, then assign a value to it
36     sumOrAvg = 0;
37
38     //for loop using index to add the balances to the sum
39     for (index = 0; index <= AR_SIZE; index++)
40     {
41         sum += balanceAr[index];
42     } //end of for loop
43
44     //calculating average
45     avg = sum / AR_SIZE;
46
47     //if statement for option 3, assigns sum to sumOrAvg
48     if (option == 3)
49     {
50         sumOrAvg = sum;
51     } //end of if statement
52
53     //if statement for option 4, assigns average to sumOrAvg
54     else if (option == 4)
55     {
```

Sum0rAvg.cpp

```
56     sum0rAvg = avg;
57 }//end of if statement
58
59 //returns a float variable and assigns it to a float variable in main
60 return sum0rAvg;
61 }
62
```

SearchName.cpp

```
1 /*****
2  * PROGRAMMER : Ali Eshghi
3  * STUDENT ID : 1112261
4  * CLASS      : CS1B
5  * SECTION    : MW 7:30pm
6  * Assign #1  : Functions and arrays
7  * DUE DATE   : 19 September 2019
8  *****/
9 #include "MyHeader.h"
10
11 /*****
12  * Function : SearchName
13  * -----
14  * This function prompts the user that who is the user searching for and uses
15  * while loop and an boolean expression to check the name array and if the name
16  * was found, returns the index of the array to be used in the parallel arrays
17  * and if the name was not found, lets the user know
18  *****/
19
20 int SearchName(string nameAr[],const int AR_SIZE)
21 {
22
23     /*****
24      * VARIABLES *
25      *****/
26
27     int index;          // PROCESS – used for the for loop
28     string searchName;  // IN & PROCESS – user input and used for the search
29     bool searchStat;    // PROCESS – changes if the name was found
30
31     //initializing the boolean expression to true
32     searchStat = true;
33
34     //initializing the index to zero to be used in the while loop
35     index = 0;
36
37     //Prompts the user for for who is he/she searching for.
38     cin.ignore(1000,'\n');
39     cout << "\nWho do you want to search for (enter done to exit) : ";
40     getline(cin,searchName);
41
42
43     //uses the while loop to check if the name was found and
44     //if the index has reached the array length.
45     while (searchStat && index <= AR_SIZE)
46     {
47         //if statement to see if the name assigned to the index of the
48         //array matches the searched name
49         if (searchName == nameAr[index])
50         {
51             //changes when the name is found
52             searchStat = false;
53         } //End of if statement
54
55         //else statement for if the name was not match with the array,
```

SearchName.cpp

```
56         // element index, then increments the index.
57         else
58         {
59             index++;
60         } //end of else statement
61
62     } //end of while loop
63
64
65     //if statement to print out the not found name and lets the user know that
66     //there is no matching name in the list with the searched name.
67     if (searchStat == true)
68     {
69         cout << searchName << " was not found." << endl << endl;
70         index = 11;
71     } //end of if statement
72
73     //else statement to let the user know that the name was found
74     else if (searchStat == false)
75     {
76         cout << "Found." << endl << endl;
77     } //end of else statement
78
79
80     //returns an integer type variable and assign it to an integer variable in the
    main.
81     return index;
82 }
83
84
```

PrintHeader.cpp

```
1 /*****
2  * PROGRAMMER : Ali Eshghi
3  * STUDENT ID : 1112261
4  * CLASS      : CS1B
5  * SECTION    : MW 7:30pm
6  * Assign #1  : Functions and arrays
7  * DUE DATE   : 19 September 2019
8  *****/
9 #include "MyHeader.h"
10
11 /*****
12  * Function : PrintHeader
13  * -----
14  * This function stores the header file into a string variable and returns that
15  * string variable to a string variable in the main called "header"
16  *****/
17
18 string PrintHeader(const string MY_NAME,
19                   const string CLASS,
20                   const string CLASS_TIME,
21                   const int    ASSIGN_NUM,
22                   const string ASSIGN_NAME)
23 {
24     /*****
25     * VARIABLES *
26     *****/
27
28     ostringstream output; //PROCESS - stores the header file as string
29
30     //This statements stores the header file into the string type variable.
31     output << left;
32     output << "*****\n" ;
33     output << "* PROGRAMMED BY : " << MY_NAME ;
34     output << "\n* " << setw(14) << "CLASS" << ": " << CLASS ;
35     output << "\n* " << setw(14) << "SECTION" << ": " << CLASS_TIME ;
36     output << "\n* LAB #" << setw(9) << ASSIGN_NUM << ": " << ASSIGN_NAME;
37     output << "\n*****\n\n" ;
38     output << right;
39
40
41     //returns the string type variable into the main
42     return output.str();
43 }
44
45
```

inFile.txt

```
1 Jean Rousseau
2 1001 15.50
3 Steve Woolston
4 1002 1423.20
5 Michele Rousseau
6 1005 52.75
7 Pete McBride
8 1007 500.32
9 Florence Rousseau
10 1010 1323.33
11 Lisa Covi
12 1009 332.35
13 Don McBride
14 1003 12.32
15 Chris Carroll
16 1008 32.35
17 Yolanda Agredano
18 1004 356.00
19 Sally Sleeper
20 1006 32.36
```