

```
1 *****
2 *   PROGRAMMED BY : Andrew Gharios
3 *   STUDENT ID    : 1449366
4 *   CLASS         : M-Th 5-7:20p
5 *   ASSIGNMENT #1 : Functions and Arrays
6 *****
7 What input file would you like to use? InFile.txt
8 What output file would you like to use? OFile.txt
9 MENU OPTIONS
10
11 1 - Find the larger balance
12 2 - Find the smaller balance
13 3 - Obtain the sum of all balances
14 4 - Obtain the average of all balances
15 5 - Find person
16 0 - Exit
17 Enter an option (0 to exit): 1
18
19 Finding the larger balance...
20
21 <redisplay menu>
22 Enter an option (0 to exit): 2
23
24 Finding the smaller balance...
25
26 <redisplay menu>
27 Enter an option (0 to exit): 3
28
29 Obtaining the sum of all balances...
30
31 <redisplay menu>
32 Enter an option (0 to exit): 4
33
34 Obtaining the average of all balances...
35
36 <redisplay menu>
37 Enter an option (0 to exit): 5
38
39 What do you want to search for (enter done to exit): Steve Woolston
40 Found.
41
42 <redisplay menu>
43 Enter an option (0 to exit): 5
44
45 What do you want to search for (enter done to exit): Jacques Rousseau
46 Jacques Rousseau was not found.
47
48 <redisplay menu>
49 Enter an option (0 to exit): 5
```

```
50
51 What do you want to search for (enter done to exit): Chris Carroll
52 Found.
53
54 <redisplay menu>
55 Enter an option (0 to exit): 5
56
57 What do you want to search for (enter done to exit): Pete McBride
58 Found.
59
60 <redisplay menu>
61 Enter an option (0 to exit): 5
62
63 What do you want to search for (enter done to exit): Jean Rousseau
64 Found.
65
66 <redisplay menu>
67 Enter an option (0 to exit): 5
68
69 What do you want to search for (enter done to exit): Florence Cyr
70 Florence Cyr was not found.
71
72 <redisplay menu>
73 Enter an option (0 to exit): 0
74
75 Thank you for using my program.
```

```
1 *****
2 *   PROGRAMMED BY : Andrew Gharios
3 *   STUDENT ID    : 1449366
4 *   CLASS         : M-Th 5-7:20p
5 *   ASSIGNMENT #1 : Functions and Arrays
6 *****
7 Larger Balance:
8 ID #    NAME                                BALANCE DUE
9 ----    -
10 1002    Steve Woolston                    $   1423.20
11
12 Smaller Balance:
13 ID #    NAME                                BALANCE DUE
14 ----    -
15 1003    Don McBride                       $    12.32
16
17 Sum of Balance for all persons:
18 $   4080.48
19
20 Average of Balance for all persons:
21 $    408.05
22
23 Search Name:
24 ID #    NAME                                BALANCE DUE
25 ----    -
26 1002    Steve Woolston                    $   1423.20
27
28 Search Name:
29 ID #    NAME                                BALANCE DUE
30 ----    -
31 1008    Chris Carroll                     $    32.35
32
33 Search Name:
34 ID #    NAME                                BALANCE DUE
35 ----    -
36 1007    Pete McBride                      $   500.32
37
38 Search Name:
39 ID #    NAME                                BALANCE DUE
40 ----    -
41 1001    Jean Rousseau                     $    15.50
42
43
```

```
1 /
    *****
    *****
2 * AUTHOR      : Andrew Gharios
3 * STUDENT ID  : 1449366
4 * AS #1      : Functions and Arrays
5 * CLASS      : CS1B
6 * SECTION    : M-TH: 5-7:20p
7 * DUE DATE   : 6/15/21
8 *****
    **/
9 #ifndef HEADER_H_
10 #define HEADER_H_
11
12 #include <iostream> // cin, cout.
13 #include <string> // string datatype variables.
14 #include <fstream> // Fstream files.
15 #include <iomanip> // fixed, setw, setprecision.
16 #include <ostream> // Ostream data type.
17 using namespace std;
18
19 /
    *****
    *****
20 * File To Array
21 * This function will take all the data from the Input file and place them
22 * into their respective Array with the corresponding datatype.
23 * ==> returns nothing
24 *
25 *****
    **/
26 void FileToArray(ifstream& InFile,
27                 const int AR_SIZE,
28                 string ArrayName[], // IN & OUT - Array of the names.
29                 int ArrayID[], // IN & OUT - Array of the Ids.
30                 float ArrayBal[]); // IN & OUT - Array of the balances.
31
32 /
    *****
    *****
33 * Search Balance
34 * This function will search for the smaller or the larger balance in the
35 * balances Array based on user selection. Then output the person and all
    their
36 * information with the smaller or larger balance.
37 * ==> returns nothing
38 *
39 *****
    **/
40 void SearchBalance(ofstream& OFile,
41                   const int AR_SIZE,
```

```

42         const float    ArrayBal[],
43         const int      ArrayIds[],
44         const string    ArrayName[], // IN & OUT - Array of the
        balances.
45         bool           larger);
46
47 /
        *****
        ****
48 * Search Name
49 * This function will search for a name within the names Array and then
        output
50 * the all the information of that found name to an output file.
51 * ==> returns nothing
52 *
53 *****
        **/
54 void SearchName(ofstream& OFile,
55                 const int AR_SIZE,
56                 string ArrayName[],
57                 int ArrayID[],
58                 float ArrayBal[]); // IN & OUT - Array of the balances.
59
60 /
        *****
        ****
61 * Sum Avg.
62 * This function will calculate the sum or the average of all the balances
        in
63 * passed array.
64 * ==> returns nothing
65 *
66 *****
        **/
67 void SumAvg(ofstream& OFile,
68             const int AR_SIZE,
69             float ArrayBal[], // IN & OUT - Array of the balances.
70             bool average);
71
72 /
        *****
        ****
73 * PrintHeaderFile
74 * This function will output the header information
75 *
76 *****
        **/
77 void PrintHeaderFile(ostream& output, // IN - output datatype.
78                     ofstream& OFile, // IN - Output to File variable.
79                     string asName, // IN - assignment name

```

```
80         int asNum,           // IN - assignment number
81         string studentName,   // IN - student's name
82         string classInfo,     // IN - class that is being taken
83         char asType,          // IN - assignment type
84         long long studentID); // IN - student ID
85
86 #endif
87
```

```
1 /
   *****
   ***
2 * AUTHOR      : Andrew Gharios
3 * STUDENT ID  : 1449366
4 * AS #1       : Functions and Arrays
5 * CLASS       : CS1B
6 * SECTION     : M-TH: 5-7:20p
7 * DUE DATE    : 6/15/21
8 *****
   */
9 #include "Header.h"
10
11 /
   *****
   ***
12 * Functions and Arrays
13 *-----
   -
14 * This program will take in which files user wants to use and then prompt a
15 * series of option for the user to be able to use the data in the file for
16 * specific tasks. The options include searching for a name within the list,
17 * comparing the balances and determining the largest or smallest, calculating
18 * the total and average of all balances. The program outputs a list at to the
19 * OFile with all request information.
20 *-----
   -
21 * INPUT:
22 * UserIfFile - User's selected InFile.
23 * UserOfFile - User's selected OFile.
24 * option     - User's selection of options.
25 *****
   */
26 int main()
27 {
28     /
   *****
   ***
29     * CONSTANTS
30     *
   -----
   -
31     * OUTPUT - USED FOR CLASS HEADING
32     *
   -----
   -
33     * PROGRAMMER : Programmer's Name
34     * CLASS      : Student's Course
35     * SECTION    : Class Days and Times
```

```

36  * LAB_NUM      : Lab Number (specific to this lab)
37  * LAB_NAME     : Title of the Lab
38  * -----
39  * INPUT_COL    : Setw size for prompt column.
40  * AR_SIZE      : Array standard size.
41  *****
    /
42
43  const string AS_NAME = "Functions and Arrays";
44  const int AS_NUM = 1;
45  const string STUDENT_NAME = "Andrew Gharios";
46  const string CLASS_INFO = "M-Th 5-7:20p";
47  const char AS_TYPE = 'A';
48  const long long STUDENT_ID = 1449366;
49
50  const int AR_SIZE = 10;
51  const int INPUT_COL = 41;
52
53  string ArNames[AR_SIZE]; // IN & CALC - Array of the names.
54  int ArIds[AR_SIZE]; // IN & CALC - Array of the ids.
55  float ArBal[AR_SIZE]; // IN & CALC - Array of the balances.
56  ifstream InFile; // IN & CALC - Input File variable.
57  ofstream OFile; // IN & CALC - Output File variable.
58  string userIfile; // IN & CALC - Input File selection from user.
59  string userOfile; // IN & CALC - Output File selection from user.
60  int option; // IN & CALC - User selection of what to do with
    data.
61  bool larger; // IN & CALC - If user wants to find the larger
    or smaller balance.
62  bool average; // IN & CALC - If user wants to calculate sum or
    average.
63
64  option = 999;
65
66  PrintHeaderFile(cout, OFile, AS_NAME, AS_NUM, STUDENT_NAME, CLASS_INFO,
67  AS_TYPE, STUDENT_ID);
68
69  cout << left;
70  cout << setw(INPUT_COL) << "What input file would you like to use?";
71  getline(cin, userIfile);
72  cout << setw(INPUT_COL) << "What output file would you like to use?";
73  getline(cin, userOfile);
74
75  InFile.open(userIfile);
76  OFile.open(userOfile);
77
78  PrintHeaderFile(OFile, OFile, AS_NAME, AS_NUM, STUDENT_NAME, CLASS_INFO,
79  AS_TYPE, STUDENT_ID);
80

```



```
81     FileToArray(InFile, AR_SIZE, ArNames, ArIds, ArBal);
82
83
84     cout << "MENU OPTIONS" << endl << endl;
85     cout << "1 - Find the larger balance" << endl;
86     cout << "2 - Find the smaller balance" << endl;
87     cout << "3 - Obtain the sum of all balances" << endl;
88     cout << "4 - Obtain the average of all balances" << endl;
89     cout << "5 - Find person" << endl;
90     cout << "0 - Exit" << endl;
91
92     while (option != 0)
93     {
94         cout << "Enter an option (0 to exit): ";
95         cin >> option;
96         cin.ignore(10000, '\n');
97
98         if (option == 1)
99         {
100             cout << endl << "Finding the larger balance...";
101             cout << endl << endl;
102             larger = true;
103             SearchBalance(OFfile, AR_SIZE, ArBal, ArIds, ArNames, larger);
104         }
105         else if (option == 2)
106         {
107             cout << endl << "Finding the smaller balance...";
108             cout << endl << endl;
109             larger = false;
110             SearchBalance(OFfile, AR_SIZE, ArBal, ArIds, ArNames, larger);
111         }
112         else if (option == 3)
113         {
114             cout << endl << "Obtaining the sum of all balances...";
115             cout << endl << endl;
116             average = false;
117             SumAvg(OFfile, AR_SIZE, ArBal, average);
118         }
119         else if (option == 4)
120         {
121             cout << endl << "Obtaining the average of all balances...";
122             cout << endl << endl;
123             average = true;
124             SumAvg(OFfile, AR_SIZE, ArBal, average);
125         }
126         else if (option == 5)
127         {
128             cout << endl;
129             SearchName(OFfile, AR_SIZE, ArNames, ArIds, ArBal);
```

```
130     }
131
132     if (option != 0)
133     {
134         cout << "<redisplay menu>" << endl;
135     }
136 }
137
138 InFile.close();
139 OFile.close();
140
141 cout << endl << "Thank you for using my program.";
142
143 return 0;
144
145 }
146
147
```

```
1 #include "Header.h"
2
3 /
4     ****
5     * File To Array
6     * This function will take all the data from the Input file and place them
7     * into their respective Array with the corresponding datatype.
8     *
9     * INPUTS:
10    * InFile    - Input File.
11    * AR_SIZE   - Array size
12    * ArrayName - Array for all the names.
13    * ArrayID   - Array for all the Ids.
14    * ArrayBal  - Array for all the balances.
15    *
16    * No Outputs.
17
18    ****
19    **/
20 void FileToArray(ifstream& InFile,      // IN - Input File variable.
21                 const int AR_SIZE,     // IN - Array sizes.
22                 string  ArrayName[],    // IN - Array for Names.
23                 int     ArrayID[],      // IN - Array for Ids.
24                 float   ArrayBal[])     // IN - Array for Balances.
25 {
26     int    index; // CALC - Index to manipulate Arrays.
27     index = 0;
28
29     while (InFile && index < AR_SIZE)
30     {
31         getline(InFile, ArrayName[index]);
32         InFile >> ArrayID[index];
33         InFile >> ArrayBal[index];
34         InFile.ignore(10000, '\n');
35         index++;
36     }
37 }
```

```
1 #include "Header.h"
2
3 /
4     *****
5     * Search Balance
6     * This function will search for the smaller or the larger balance in the
7     * balances Array based on user selection. Then output the person and all
8     * their
9     * information with the smaller or larger balance.
10    *
11    * INPUTS:
12    * OFile      - Output File.
13    * AR_SIZE    - Array size
14    * ArrayBal   - Array for all the balances.
15    * ArrayName  - Array for all the names.
16    * ArrayID    - Array for all the Ids.
17    * average    - Bool to calculate avg or sum.
18    *
19    * No Outputs.
20    *****
21    ***/
22 void SearchBalance(ofstream& OFile,          // IN - Output File
23                   variable,
24                   const int AR_SIZE,         // IN - Array sizes.
25                   const float ArrayBal[],    // IN - Array for Balances.
26                   const int ArrayIds[],      // IN - Array for Ids.
27                   const string ArrayName[],  // IN - Array for Names.
28                   bool larger)              // IN - bool to calculate
29                                           larger or smaller bal.
30 {
31     const int ID_COL = 9;    // CALC - Setw size for ID column.
32     const int NAME_COL = 25; // CALC - Setw size for Name column.
33     const int BAL_COL = 10;  // CALC - Setw size for Balance column.
34
35     int index;    // CALC - Index to manipulate Arrays.
36     float largerBal; // CALC - Largest balance storage.
37     float smallerBal; // CALC - Smallest balance storage.
38     int location;  // CALC - Location to store index for output.
39
40     index      = 0;
41     largerBal  = 0;
42     location   = 0;
43     smallerBal = 10000;
44
45     while (index < AR_SIZE)
46     {
47         if (larger)
48         {
49             if (largerBal < ArrayBal[index])
50             {
51                 largerBal = ArrayBal[index];
52             }
53         }
54     }
55 }
```

```
48         location = index;
49     }
50 }
51 else
52 {
53     if (smallerBal > ArrayBal[index])
54     {
55         smallerBal = ArrayBal[index];
56         location = index;
57     }
58 }
59 index++;
60 }
61
62
63 OFile << left;
64 OFile << fixed;
65 OFile << setprecision(2);
66
67 if (larger)
68 {
69     OFile << "Larger Balance:" << endl;
70     OFile << setw(ID_COL) << "ID #" << setw(NAME_COL);
71     OFile << "NAME" << "BALANCE DUE" << endl;
72
73     OFile << setw(ID_COL) << "----";
74     OFile << setw(NAME_COL) << "-----";
75     OFile << "-----" << endl;
76
77     OFile << setw(ID_COL) << ArrayIds[location];
78     OFile << setw(NAME_COL) << ArrayName[location];
79     OFile << "$" << right << setw(BAL_COL) << ArrayBal[location];
80     OFile << endl << endl;
81     OFile << setprecision(6);
82 }
83 else
84 {
85     OFile << "Smaller Balance:" << endl;
86     OFile << setw(ID_COL) << "ID #" << setw(NAME_COL);
87     OFile << "NAME" << "BALANCE DUE" << endl;
88
89     OFile << setw(ID_COL) << "----";
90     OFile << setw(NAME_COL) << "-----";
91     OFile << "-----" << endl;
92
93     OFile << setw(ID_COL) << ArrayIds[location];
94     OFile << setw(NAME_COL) << ArrayName[location];
95     OFile << "$" << right << setw(BAL_COL) << ArrayBal[location];
96     OFile << endl << endl;
97     OFile << setprecision(6);
98 }
99 }
100
```

```

1  #include "Header.h"
2
3  /
    *****
    ****
4  * Search Name
5  *   This function will search for a name within the names Array and then
    output
6  *   the all the information of that found name to an output file.
7  *
8  * INPUTS:
9  *   OFile - Output File.
10 *   AR_SIZE - Array size
11 *   ArrayBal - Array for all the balances.
12 *   ArrayName - Array for all the names.
13 *   ArrayID - Array for all the Ids.
14 *   average - Bool to calculate avg or sum.
15 *
16 *No Outputs.
17 *
    *****
    ***/
18 void SearchName(ofstream& OFile, // IN - Output File variable.
19                 const int AR_SIZE, // IN - Array sizes.
20                 string ArrayName[], // IN - Array for Names.
21                 int ArrayID[], // IN - Array for Ids.
22                 float ArrayBal[]) // IN - Array for Balances.
23 {
24     const int ID_COL = 9; // CALC - Setw size for ID column.
25     const int NAME_COL = 25; // CALC - Setw size for Name column.
26     const int BAL_COL = 10; // CALC - Setw size for Balance column.
27
28     int index; // CALC - Index to manipulate Arrays.
29     string userPick; // IN & CALC - User's pick for which name to search.
30     bool found; // CALC - If name was found or not.
31
32     found = false;
33     index = 0;
34
35     cout << "What do you want to search for (enter done to exit): ";
36     getline(cin, userPick);
37
38     while (index < AR_SIZE && !found)
39     {
40         if (userPick == ArrayName[index])
41         {
42             found = true;
43             cout << "Found." << endl << endl;
44
45             OFile << left;
46             OFile << fixed;
47             OFile << setprecision(2);
48

```

```
49         OFile << "Search Name:" << endl;
50         OFile << setw(ID_COL) << "ID #" << setw(NAME_COL);
51         OFile << "NAME" << "BALANCE DUE" << endl;
52
53         OFile << setw(ID_COL) << "----";
54         OFile << setw(NAME_COL) << "-----";
55         OFile << "-----" << endl;
56
57         OFile << setw(ID_COL) << ArrayID[index];
58         OFile << setw(NAME_COL) << ArrayName[index];
59         OFile << "$" << right << setw(BAL_COL) << ArrayBal[index];
60         OFile << endl << endl;
61     }
62     else
63     {
64         index++;
65     }
66 }
67 if (!found)
68 {
69     cout << userPick << " was not found." << endl << endl;
70 }
71 }
72
```

```
1 #include "Header.h"
2
3
4 /
5 * Sum Avg.
6 * This function will calculate the sum or the average of all the balances
  in
7 * passed array.
8 *
9 * INPUTS:
10 * OFile - Output File.
11 * AR_SIZE - Array size
12 * ArrayBal - Array for all the balances.
13 * average - Bool to calculate avg or sum.
14 *
15 * No Outputs.
16 ****
17 void SumAvg(ofstream& OFile, // IN - Output File variable.
18             const int AR_SIZE, // IN - Array sizes.
19             float ArrayBal[], // IN - Array for Balances.
20             bool average) // IN - Bool to calculate avg or sum.
21 {
22     int index; // CALC - Index to manipulate Arrays.
23     float totalBal; // CALC & OUT - Sum of all balances.
24     float avg; // CALC & OUT - Avg of all balances.
25
26     totalBal = 0;
27
28     for (index = 0; index < AR_SIZE; index++)
29     {
30         totalBal += ArrayBal[index];
31     }
32
33     OFile << setprecision(2);
34     OFile << fixed;
35
36     if (average)
37     {
38         avg = totalBal / AR_SIZE;
39         OFile << "Average of Balance for all persons:" << endl;
40         OFile << "$" << setw(9) << avg;
41         OFile << endl << endl;
42     }
43     else
44     {
45
46         OFile << "Sum of Balance for all persons:" << endl;
47         OFile << "$" << setw(9) << totalBal;
48         OFile << endl << endl;
49     }
```



```
50     OFile << setprecision(6);  
51 }
```

```

1  #include "Header.h"
2
3  /
    *****
    ****
4  * PrintHeaderFile
5  *   This function will output the header information
6
    *
    -----
7  * PRE-CONDITIONS
8  *   The following parameters need to have a defined value prior to calling
9  *   the function
10 *
11 *       asName: The name of the assignment given in the course
12 *       asNum: The number of the assignment given in the course
13 *       studentName: The name of the student writing the code
14 *       classInfo: The course name, date, and time of the class
15 *       asType: Will either output as a lab or an assignment
16 *       studentID: The Identification Number of the student
    *****
    **/
17
18 void PrintHeaderFile(ostream& output,      // IN - output datatype.
19                     ofstream& OFile,      // IN - Output to File variable.
20                     string asName,         // IN - assignment name
21                     int asNum,            // IN - assignment number
22                     string studentName,   // IN - student's name
23                     string classInfo,     // IN - class that is being taken
24                     char asType,          // IN - assignment type
25                     long long studentID) // IN - student ID
26 {
27     output << left;
28     output <<
        "*****\n";
29     output << "*" PROGRAMMED BY : " << studentName << endl;
30     output << "*" " << setw(14) << "STUDENT ID " << ": " << studentID <<
        endl;
31     output << "*" " << setw(14) << "CLASS " << ": " << classInfo <<
        endl;
32     output << "*" ";
33
34     // PROCESSING - This will adjust setws and format appropriately based
35     //               on if this is a lab 'L' or assignment
36
37     if (toupper(asType) == 'L')
38     {
39         output << "LAB #" << setw(9);
40     }
41     else
42     {
43         output << "ASSIGNMENT #" << setw(2);
44     }

```

```
45     output << asNum << ": " << asName << endl;
46     output <<
        "*****";
47     output << right << endl;
48
49     return;
50 }
```

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