```
 1  ***********************************************************
 2  *    PROGRAMMED BY : Andrew Gharios
 3  *    STUDENT ID    : 1449366
 4  *    CLASS         : M-Th 5-7:20p
 5  *    LAB #6        : Structs
 6  ************************************************************
 7  What input file would you like to use?   InFile.txt
 8  What output file would you like to use?  OutFile.txt
 9
10  What do you want to search for (enter done to exit): Steve Woolston
11  Found.
12
13  What do you want to search for (enter done to exit): Jacques Rousseau
14  Jacques Rousseau was not found.
15
16  What do you want to search for (enter done to exit): Chris Carroll
17  Found.
18
19  What do you want to search for (enter done to exit): Lisa Covi
20  Found.
21
22  What do you want to search for (enter done to exit): Florence Rousseau
23  Found.
24
25  What do you want to search for (enter done to exit): Frankie Lane
26  Frankie Lane was not found.
27
28  What do you want to search for (enter done to exit): done
29
30  Thank you for using my program.
```

```
 1  *********************************************************
 2  *    PROGRAMMED BY : Andrew Gharios
 3  *    STUDENT ID    : 1449366
 4  *    CLASS         : M-Th 5-7:20p
 5  *    LAB #6        : Structs
 6  *********************************************************
 7  Larger Balance:
 8  ID #      NAME                  BALANCE DUE
 9  ----      --------------------  -----------
10  1002      Steve Woolston        $   1423.20
11
12  Sum of Balance for all persons:
13  $   4080.48
14
15  Search Name:
16  ID #      NAME                  BALANCE DUE
17  ----      --------------------  -----------
18  1002      Steve Woolston        $   1423.20
19
20  Search Name:
21  ID #      NAME                  BALANCE DUE
22  ----      --------------------  -----------
23  1008      Chris Carroll         $     32.35
24
25  Search Name:
26  ID #      NAME                  BALANCE DUE
27  ----      --------------------  -----------
28  1009      Lisa Covi             $    332.35
29
30  Search Name:
31  ID #      NAME                  BALANCE DUE
32  ----      --------------------  -----------
33  1010      Florence Rousseau     $   1323.33
34
35
```

```cpp
 1  /
    *************************************************************************
    **
 2  * AUTHOR      : Andrew Gharios
 3  * STUDENT ID : 1449366
 4  * LAB #6      : Structs
 5  * CLASS       : CS1B
 6  * SECTION     : M-TH: 5-7:20p
 7  * DUE DATE    : 6/26/21
 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  struct Balance
20  {
21      string name;
22      int    id;
23      float  bal;
24  };
25
26  /*************************************************************************
27   * File To Array
28   *  This function will take all the data from the Input file and place them
29   *  into their respective Array with the corresponding datatype.
30   *  ==> returns nothing
31
    *************************************************************************/
32  void FileToArray(ifstream& InFile,          // IN - Input File variable.
33                   const int AR_SIZE,         // IN - Array sizes.
34                   Balance   ArrayAccount[]); // IN - Array for accounts.
35
36  /*************************************************************************
37   * Search Balance
38   *  This function will search for the larger balance in the
39   *  balances Array based on user selection.
40   *  ==> returns location of largest balance.
41
    *************************************************************************/
42  int SearchBalance(const int     AR_SIZE, // IN - Array sizes.
43                    Balance ArrayAcc[]); // IN - Array for accounts.
44
```

```
45  /**************************************************************************
46   * Search Name
47   *  This function will search for a name within the names Array and then output
48   *  the all the information of that found name to an output file.
49   *  ==> returns nothing
50                                                                              ↵
     **************************************************************************/
51  void SearchName(ofstream& OFile,  // IN - Output File variable.
52                  const int AR_SIZE,    // IN - Array sizes.
53                  Balance   ArrayAcc[], // IN - Array for accounts.
54                  string    userPick);  // IN - User's name pick for searching.
55
56  /**************************************************************************
57   * Sum Avg.
58   *  This function will calculate the sum or the average of all the balances in
59   *  passed array.
60   *  ==> returns nothing
61                                                                              ↵
     **************************************************************************/
62  void SumAr(ofstream& OFile,
63             const int AR_SIZE,
64             Balance   ArrayAcc[]); // IN & OUT - Array of the balances.
65
66  /**************************************************************************
67   * PrintHeaderFile
68   *   This function will output the header information
69                                                                              ↵
     **************************************************************************/
70  void PrintHeaderFile(ostream& output,      // IN - output datatype.
71     ofstream& OFile,      // IN - Output to File variable.
72     string asName,        // IN - assignment name
73     int asNum,             // IN - assignment number
74     string studentName,   // IN - student's name
75     string classInfo,     // IN - class that is being taken
76     char asType,          // IN - assignment type
77     long long studentID); // IN - student ID
78
79  #endif
80
81
```

```cpp
 1  /
    ***************************************************************************
    ***
 2  * AUTHOR     : Andrew Gharios
 3  * STUDENT ID : 1449366
 4  * LAB #6     : Structs
 5  * CLASS      : CS1B
 6  * SECTION    : M-TH: 5-7:20p
 7  * DUE DATE   : 6/26/21
 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  * userPick   - User's pick for searching a name.
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      * ID_COL    : Setw size for ID column.
42      * NAME_COL  : Setw size for name column.
43      * BAL_COL   : Setw size for balance column.
44      **************************************************************************** ⮠
          /
45
46      const string AS_NAME = "Structs";
47      const int AS_NUM = 6;
48      const string STUDENT_NAME = "Andrew Gharios";
49      const string CLASS_INFO = "M-Th 5-7:20p";
50      const char AS_TYPE = 'L';
51      const long long STUDENT_ID = 1449366;
52
53      const int AR_SIZE = 10;
54      const int INPUT_COL = 41;
55      const int ID_COL = 9;
56      const int NAME_COL = 25;
57      const int BAL_COL = 10;
58
59      Balance  ArAcc[AR_SIZE];   // IN & CALC - Array of the accounts.
60      ifstream InFile;       // IN & CALC - Input File variable.
61      ofstream OFile;            // IN & CALC - Output File variable.
62      string   userPick;         // IN & CALC - User's pick for which name to   ⮠
          search.
63      string   userIfile;        // IN & CALC - Input File selection from user.
64      string   userOfile;        // IN & CALC - Output File selection from user.
65      int      location;         // CALC      - Location of largest balance.
66
67      PrintHeaderFile(cout, OFile, AS_NAME, AS_NUM, STUDENT_NAME, CLASS_INFO,
68          AS_TYPE, STUDENT_ID);
69
70      cout << left;
71      cout << setw(INPUT_COL) << "What input file would you like to use?";
72      getline(cin, userIfile);
73      cout << setw(INPUT_COL) << "What output file would you like to use?";
74      getline(cin, userOfile);
75      cout << endl;
76
77      InFile.open(userIfile);
78      OFile.open(userOfile);
79
80      PrintHeaderFile(OFile, OFile, AS_NAME, AS_NUM, STUDENT_NAME, CLASS_INFO,
81          AS_TYPE, STUDENT_ID);
82
```

```
83          FileToArray(InFile, AR_SIZE, ArAcc);
84
85          location = SearchBalance(AR_SIZE, ArAcc);
86
87          OFile << left;
88          OFile << fixed;
89          OFile << setprecision(2);
90
91          OFile << "Larger Balance:" << endl;
92          OFile << setw(ID_COL) << "ID #" << setw(NAME_COL);
93          OFile << "NAME" << "BALANCE DUE" << endl;
94
95          OFile << setw(ID_COL) << "----";
96          OFile << setw(NAME_COL) << "--------------------";
97          OFile << "-----------" << endl;
98
99          OFile << setw(ID_COL)   << ArAcc[location].id;
100         OFile << setw(NAME_COL) << ArAcc[location].name;
101         OFile << "$"   << right   << setw(BAL_COL) << ArAcc[location].bal;
102         OFile << endl << endl;
103         OFile << setprecision(6);
104
105         SumAr(OFile, AR_SIZE, ArAcc);
106
107         cout << "What do you want to search for (enter done to exit): ";
108         getline(cin, userPick);
109
110         while (userPick != "done")
111         {
112             SearchName(OFile, AR_SIZE, ArAcc, userPick);
113
114             cout << "What do you want to search for (enter done to exit): ";
115             getline(cin, userPick);
116         }
117
118         InFile.close();
119         OFile.close();
120
121         cout << endl << "Thank you for using my program.";
122
123         return 0;
124     }
125
```

```cpp
 1  #include "Header.h"
 2
 3  /*****************************************************************************
 4   * Search Balance
 5   *  This function will search for the smaller or the larger balance in the
 6   *  balances Array based on user selection and return it's location.
 7   *
 8   * INPUTS:
 9   *  OFile     - Output File.
10   *  AR_SIZE   - Array size.
11   *  ArrayAcc  - Array for all the accounts.
12   *
13   * No Outputs.
14   *****************************************************************************/
15  int SearchBalance(const int     AR_SIZE,// IN - Array sizes.
16                    Balance ArrayAcc[]) // IN - Array for accounts.
17  {
18      int    index;      // CALC - Index to manipulate Arrays.
19      int    location;   // CALC - Location of largest balance.
20      float  largerBal;  // CALC - Largest balance storage.
21
22      index = 0;
23      largerBal = 0;
24
25      while (index < AR_SIZE)
26      {
27          if (largerBal < ArrayAcc[index].bal)
28          {
29              largerBal = ArrayAcc[index].bal;
30              location  = index;
31          }
32          index++;
33      }
34
35      return location;
36
37
38
39  }
40
```

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

```cpp
 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   *  ArrayAcc - Array for all the accounts.
12   *
13  *No Outputs.
14   *                                                                          ⮑
     ************************************************************************** ⮑
     /
15  void SearchName(ofstream& OFile,  // IN - Output File variable.
16                  const int AR_SIZE,    // IN - Array sizes.
17                  Balance   ArrayAcc[], // IN - Array for accounts.
18                  string    userPick)   // IN - User's name pick for searching.
19  {
20      const int ID_COL = 9;     // CALC - Setw size for ID column.
21      const int NAME_COL = 25;  // CALC - Setw size for Name column.
22      const int BAL_COL = 10;   // CALC - Setw size for Balance column.
23
24      int     index;    // CALC     - Index to manipulate Arrays.
25      bool found;    // CALC      - If name was found or not.
26
27      found = false;
28      index = 0;
29
30      while (index < AR_SIZE && !found)
31      {
32          if (userPick == ArrayAcc[index].name)
33          {
34              found = true;
35              cout << "Found." << endl << endl;
36
37              OFile << left;
38              OFile << fixed;
39              OFile << setprecision(2);
40
41              OFile << "Search Name:" << endl;
42              OFile << setw(ID_COL) << "ID #" << setw(NAME_COL);
43              OFile << "NAME" << "BALANCE DUE" << endl;
44
45              OFile << setw(ID_COL) << "----";
46              OFile << setw(NAME_COL) << "--------------------";
47              OFile << "-----------" << endl;
```

```
48
49                 OFile << setw(ID_COL) << ArrayAcc[index].id;
50                 OFile << setw(NAME_COL) << ArrayAcc[index].name;
51                 OFile << "$" << right << setw(BAL_COL) << ArrayAcc[index].bal;
52                 OFile << endl << endl;
53             }
54         else
55         {
56             index++;
57         }
58     }
59     if (!found)
60     {
61         cout << userPick << " was not found." << endl << endl;
62     }
63 }
64
```

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

```cpp
 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   *          asName: The name of the assignment given in the course
11   *          asNum: The number of the assignment given in the course
12   *          studentName: The name of the student writing the code
13   *          classInfo: The course name, date, and time of the class
14   *          asType: Will either output as a lab or an assignment
15   *          studentID: The Identification Number of the student
16   *
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
```