```
1
     #include <iostream>
 2
     #include <string>
 3
     #include <fstream>
     #include <iomanip>
 4
 5
 6
     using namespace std;
 7
 8
     const int MaxList = 200;
 9
     struct listtype {
10
11
     string last;
12
     string first;
13
     int id;
14
15
16
     const listtype initpositions = { "LastName" , "FirstName" , 0 };
17
18
     void PrintTopTitles(ofstream &outf) {
19
     outf << setw(13) << setfill(' ') << right << " " << "CSCI 112" << endl;
     outf << setw(12) << setfill (' ') << right << " " << "Class List" <<endl;
20
21
     outf << setw(35) << setfill('-') << right << "- " << endl;
22
     outf << setw(6) << setfill(' ') << left << "ID" << " " << "Name" << endl;
23
     outf << setw(35) << setfill('-') << right << "- " << endl;
24
25
26
     void PrintBottomTitles(ofstream &outf) {
27
     outf << setw(35) << setfill('-') << right << "- " << endl;
28
     }
29
     void CreateList(listtype List[], int &numlist) {
30
31
     for (int i=0; i < MaxList; i++) List[i] = initpositions;</pre>
32
     numlist = 0;
33
34
35
     void ReadData(listtype List[], int &numlist) {
36
     ifstream inf;
37
     inf.open("program3datafile.txt");
38
     int curnum = 0;
39
     int position = -1;
40
     while (!inf.eof()) {
41
         inf >> List[curnum].last >> List[curnum].first >> List[curnum].id >> ws;
42
         position = List[curnum].last.find(",");
43
             if (position >=0) List[curnum].last.erase(position,1);
44
         curnum++;
45
         }
     numlist = curnum;
46
47
48
49
     bool ListisFull(int numlist) {
50
     return (numlist == MaxList);
51
     }
52
53
     bool ListisEmpty(int numlist) {
     return (numlist = 0);
54
```

```
55
      }
 56
 57
      void swap(listtype &x, listtype &y) {
 58
      listtype temp;
 59
      temp = x;
 60
      x=y;
 61
      y = temp;
 62
 63
 64
      void sortbyalpha(listtype List[], int numlist) {
 65
      for (int i = 0; i< numlist -1; i++) {</pre>
 66
          for (int j = 0; j < numlist - 1; j++) {</pre>
 67
              if (List[j].last > List[j+1].last) {
 68
                   swap(List[j], List[j+1]); } }
 69
      }
 70
 71
      void TraverseList(listtype List[], ofstream &outf,int numlist) {
 72
          string fullname;
 73
          if (!ListisEmpty(numlist)) {
 74
              PrintTopTitles(outf);
 75
              sortbyalpha(List, numlist);
 76
              for (int i=0; i<numlist; i++) {</pre>
 77
                   fullname = List[i].first + " " + List[i].last;
                   outf << setw(6) << setfill(' ') << left << List[i].id << " " << fullname
 78
                                                                                                  7
                   << endl;
 79
 80
              PrintBottomTitles(outf);
 81
 82
          else outf << "The list is empty" << endl;</pre>
 83
      }
 84
 85
      void InsertList(listtype List[], int &numlist, ofstream &outf,string last, string
      first, int id) {
 86
      int destination;
 87
      if (!ListisFull(numlist)) {
 88
          destination = 0;
 89
          while (destination <= numlist && last >= List[destination].last) {
 90
              destination++; }
          for (int i=numlist-1; i>=destination; i--) List[i+1]=List[i];
 91
 92
          List[destination].last = last;
 93
          List[destination].first = first;
 94
          List[destination].id = id;
 95
          numlist++;}
      else outf<< "The list is full" << endl;</pre>
 96
 97
      }
 98
 99
100
      void DeletefromList(listtype List[], string last, string first, int id, ofstream
                                                                                                  ₽
      &outf, int &numlist) {
      int destination=0;
101
102
      int i=0;
103
      while (last != List[i].last || first != List[i].first || id != List[i].id) { i++;
                                                                                                  7
      destination++;}
      if (last == List[destination].last && first == List[destination].first && id ==
                                                                                                  ₽
104
```

C:\Users\brynl\Documents\Semester2\ComputerScience2\Program3ExtraCredit\program3sourcecode.cpp Page 3 of 3 2/7/2018 5:42:22 PM

```
List[destination].id) {
105
          for (int w=destination; w<numlist; w++) {</pre>
106
                  List[w] = List[w+1];
107
                  }
108
              List[numlist-1]=initpositions;
109
              numlist--;
110
              }
      else outf << "Entry, " << first << " " << last << ", ID number " << id << " was not
111
      found in this list." << endl;</pre>
112
      }
113
114
      void PrintSpaces(ofstream &outf) {
115
      outf << endl << endl << endl;
116
117
118
      int main(){
119
      listtype List[MaxList];
120
      int numlist;
121
      ofstream outf;
122
      outf.open("program3outputfile.txt");
      CreateList(List, numlist);
123
124
      ReadData(List, numlist);
      TraverseList(List,outf, numlist);
125
126
      PrintSpaces(outf);
127
      InsertList(List, numlist, outf, "Skywalker", "Luke", 444);
128
      TraverseList(List,outf, numlist);
129
      outf << "Luke Skywalker, ID #444, was added to the class." << endl;
130
      PrintSpaces(outf);
131
      DeletefromList(List, "Organa" , "Leia", 355, outf, numlist);
132
      TraverseList(List, outf, numlist);
133
      outf << "Leia Organa, ID #355, was deleted from the class." << endl;
134
      outf << endl;
      system("pause");
135
136
      return 0;
137
      }
138
139
```