```cpp
#include <iostream>
#include <string>
#include <fstream>
#include <iomanip>

using namespace std;

const int MaxList = 200;

struct listtype {
string last;
string first;
int id;
};

const listtype initpositions = { "LastName" , "FirstName" , 0 };

void PrintTopTitles(ofstream &outf) {
outf << setw(13) << setfill(' ') << right << " " << "CSCI 112" << endl;
outf << setw(12) << setfill (' ') << right << " " << "Class List" <<endl;
outf << setw(35) << setfill('-') << right << "- " << endl;
outf << setw(6) << setfill(' ') << left << "ID" << " " << "Name" << endl;
outf << setw(35) << setfill('-') << right << "- " << endl;
}

void PrintBottomTitles(ofstream &outf) {
outf << setw(35) << setfill('-') << right << "- " << endl;
}

void CreateList(listtype List[], int &numlist) {
for (int i=0; i < MaxList; i++) List[i] = initpositions;
numlist = 0;
}

void ReadData(listtype List[], int &numlist) {
ifstream inf;
inf.open("program3datafile.txt");
int curnum = 0;
int position = -1;
while (!inf.eof()) {
    inf >> List[curnum].last >> List[curnum].first >> List[curnum].id >> ws;
    position = List[curnum].last.find(",");
        if (position >=0) List[curnum].last.erase(position,1);
    curnum++;
    }
numlist = curnum;
}

bool ListisFull(int numlist) {
return (numlist == MaxList);
}

bool ListisEmpty(int numlist) {
return (numlist = 0);
```

```cpp
}

void swap(listtype &x, listtype &y) {
listtype temp;
temp = x;
x=y;
y = temp;
}

void sortbyalpha(listtype List[], int numlist) {
for (int i = 0; i< numlist -1; i++) {
    for (int j = 0; j < numlist - 1; j++) {
        if (List[j].last > List[j+1].last) {
            swap(List[j], List[j+1]); } } }
}

void TraverseList(listtype List[], ofstream &outf,int numlist) {
    string fullname;
    if (!ListisEmpty(numlist)) {
        PrintTopTitles(outf);
        sortbyalpha(List, numlist);
        for (int i=0; i<numlist; i++) {
            fullname = List[i].first + " " + List[i].last;
            outf << setw(6) << setfill(' ') << left << List[i].id << " " << fullname << endl;
        }
        PrintBottomTitles(outf);
    }
    else outf << "The list is empty" << endl;
}

void InsertList(listtype List[], int &numlist, ofstream &outf,string last, string first, int id) {
int destination;
if (!ListisFull(numlist)) {
    destination = 0;
    while (destination <= numlist && last >= List[destination].last) {
        destination++; }
    for (int i=numlist-1; i>=destination; i--) List[i+1]=List[i];
    List[destination].last = last;
    List[destination].first = first;
    List[destination].id = id;
    numlist++;}
else outf<< "The list is full" << endl;
}


void DeletefromList(listtype List[], string last, string first, int id, ofstream &outf, int &numlist){
int destination=0;
int i=0;
while (last != List[i].last || first != List[i].first || id != List[i].id) { i++; destination++;}
if (last == List[destination].last && first == List[destination].first && id ==
List[destination].id) {
    for (int w=destination;w<numlist;w++) {
            List[w]=List[w+1];
            }
        List[numlist-1]=initpositions;
        numlist--;
        }
else outf << "Entry, " << first << " " << last << ", ID number " << id << " was not found in this list." << endl;
}

void PrintSpaces(ofstream &outf) {
outf << endl << endl << endl << endl;
}

int main(){
listtype List[MaxList];
int numlist;
ofstream outf;
outf.open("program3outputfile.txt");
CreateList(List, numlist);
ReadData(List, numlist);
TraverseList(List,outf, numlist);
PrintSpaces(outf);
InsertList(List,numlist,outf, "Skywalker", "Luke", 444);
TraverseList(List,outf, numlist);
outf << "Luke Skywalker, ID #444, was added to the class." << endl;
PrintSpaces(outf);
DeletefromList(List, "Organa" , "Leia", 355, outf, numlist);
TraverseList(List, outf, numlist);
outf << "Leia Organa, ID #355, was deleted from the class." << endl;
outf << endl;
system("pause");
return 0;
}


```