```cpp
/*
 *  - main.cpp -
 * ENSC 251 - Lab Assignment 2
 * May 31, 2019
 * Judd Foster
 * 301377893
 */

// include standard libraries
#include <iostream>
#include <fstream>
#include <sstream>
#include <cstdlib>
// include user defined libraries
#include "student.hpp"
#include "domesticstudent.hpp"
#include "toeflscore.hpp"
#include "internationalstudent.hpp"

using namespace std;

DomesticStudent* sortBy(DomesticStudent* unsortedDomesticStudents, int num, string type,
 string order)
{
    int currentIndex;
    int ascending;

    if (type != "F" && type != "L" && type != "C" && type != "R") return NULL;//return
    -1;
    if (order == "ascending") ascending = 1;
    else ascending = -1;

    DomesticStudent* sortedDomesticStudents = new DomesticStudent[num];
    int* usedIndex = new int[num];

    for (int i = 0; i < num; i++)
    {
        for (int j = 0; j < num; j++)
        {
            if (usedIndex[j] != 1)
            {
                currentIndex = j;
                break;
            }
        }
        for (int j = 0; j < num; j++)
        {
            if (usedIndex[j] == 1) continue;
            if (((type == "F") && (compareFirstName(unsortedDomesticStudents[
            currentIndex], unsortedDomesticStudents[j]) * ascending > 0))
            ||  ((type == "L") && (compareLastName(unsortedDomesticStudents[currentIndex
            ], unsortedDomesticStudents[j]) * ascending > 0))
            ||  ((type == "C") && (compareCGPA(unsortedDomesticStudents[currentIndex],
            unsortedDomesticStudents[j]) * ascending > 0))
            ||  ((type == "R") && (compareResearchScore(unsortedDomesticStudents[
            currentIndex], unsortedDomesticStudents[j]) * ascending) > 0))
            {
                currentIndex = j;
            }
        }
        sortedDomesticStudents[i] = unsortedDomesticStudents[currentIndex];
        usedIndex[currentIndex] = 1;
    }

    delete [] usedIndex;
    return sortedDomesticStudents;
}

InternationalStudent* sortBy(InternationalStudent* unsortedInternationalStudents, int
```

```
    num, string type, string order)
64  {
65      int currentIndex;
66      int ascending;
67
68      if (type != "F" && type != "L" && type != "C" && type != "R") return NULL;
69      if (order == "ascending") ascending = 1;
70      else ascending = -1;
71
72      InternationalStudent* sortedInternationalStudents = new InternationalStudent[num];
73      int* usedIndex = new int[num];
74
75      int k = 0;
76
77      for (int i = 0; i < num; i++)
78      {
79          for (int j = 0; j < num; j++)
80          {
81              if (usedIndex[j] != 1)
82              {
83                  currentIndex = j;
84                  break;
85              }
86          }
87          for (int j = 0; j < num; j++)
88          {
89              if (usedIndex[j] == 1) continue;
90              if (((type == "F") && compareFirstName(unsortedInternationalStudents[
                currentIndex], unsortedInternationalStudents[j]) == ascending)
91              ||  ((type == "L") && compareLastName(unsortedInternationalStudents[
                currentIndex], unsortedInternationalStudents[j]) == ascending)
92              ||  ((type == "C") && compareCGPA(unsortedInternationalStudents[currentIndex
                ], unsortedInternationalStudents[j]) == ascending)
93              ||  ((type == "R") && compareResearchScore(unsortedInternationalStudents[
                currentIndex], unsortedInternationalStudents[j]) == ascending))
94              {
95                  currentIndex = j;
96              }
97          }
98          if (unsortedInternationalStudents[currentIndex].getReading() > 20
99          &&  unsortedInternationalStudents[currentIndex].getListening() > 20
100         &&  unsortedInternationalStudents[currentIndex].getSpeaking() > 20
101         &&  unsortedInternationalStudents[currentIndex].getWriting() > 20
102         &&  unsortedInternationalStudents[currentIndex].getTotalScore() > 93)
103         {
104             sortedInternationalStudents[k++] = unsortedInternationalStudents[
                currentIndex];
105         }
106         usedIndex[currentIndex] = 1;
107     }
108
109     delete [] usedIndex;
110
111     return sortedInternationalStudents;
112 }
113
114 DomesticStudent* sortBy(DomesticStudent* unsortedDomesticStudents, int num, string type)
115 {
116     return sortBy(unsortedDomesticStudents, num, type, "ascending");
117 }
118
119 InternationalStudent* sortBy(InternationalStudent* unsortedDomesticStudents, int num,
    string type)
120 {
121     return sortBy(unsortedDomesticStudents, num, type, "ascending");
122 }
123
124 ifstream fileOpen(string fname)
125 {
```

```
126         ifstream file(fname);
127         if(!file.is_open())
128         {
129             cout << "Unable to open file " << fname << ". Terminating" << endl;
130             exit(-1);
131         }
132         return file;
133     }
134
135     int countLines(ifstream& file)
136     {
137         int i = 0;
138         string line;
139         while(getline(file, line)) i++;
140         return i;
141     }
142
143
144     int main()
145     {
146         string studentSelection, sortingSelection, line;
147         cout << "Please select which group you would like to sort:\nDomestic Students (D)
            International Students(I)  Overall Sort (O)\n";
148         cin >> studentSelection;
149         cout << "Please enter what you would like to sort by:\nFirst Name (F)  Last Name
            (L)  CGPA (C)  Research Score (R)\n";
150         cin >> sortingSelection;
151
152
153
154         if (studentSelection == "D" || studentSelection == "O")
155         {
156             ifstream domesticFile = fileOpen("domestic-stu.txt");
157
158             int numDomesticStudents = countLines(domesticFile) - 1;
159
160             DomesticStudent* unsortedDomesticStudents = new DomesticStudent[
                numDomesticStudents];
161
162             domesticFile.clear();
163             domesticFile.seekg(0, ios::beg);
164             getline(domesticFile, line);
165
166             int i = 0;
167             while(getline(domesticFile, line))
168             {
169                 istringstream ss(line);
170                 ss >> unsortedDomesticStudents[i++];
171             }
172
173             DomesticStudent* sortedDomesticStudents = sortBy(unsortedDomesticStudents,
                numDomesticStudents, sortingSelection);
174
175             if (sortedDomesticStudents == NULL)
176             {
177                 cout << "Your input was invalid. Please re run this program\n";
178                 return -1;
179             }
180
181             cout << "\n\nSorted DomesticStudent Array:\n\n";
182             for (int i = 0; i < numDomesticStudents; i++) cout << sortedDomesticStudents[i];
183
184             domesticFile.close();
185             delete [] unsortedDomesticStudents;
186             delete [] sortedDomesticStudents;
187         }
188         else if (studentSelection == "I" || studentSelection == "O")
189         {
190             ifstream internationalFile = fileOpen("international-stu.txt");
```

```
191
192             int numInternationalStudents = countLines(internationalFile) - 1;
193
194             InternationalStudent* unsortedInternationalStudents = new InternationalStudent[
                numInternationalStudents];
195
196             internationalFile.clear();
197             internationalFile.seekg(0, ios::beg);
198             getline(internationalFile, line);
199
200             int i = 0;
201             while(getline(internationalFile, line))
202             {
203                 istringstream ss(line);
204                 ss >> unsortedInternationalStudents[i++];
205             }
206
207             InternationalStudent* sortedInternationalStudents = sortBy(
                unsortedInternationalStudents, numInternationalStudents, sortingSelection);
208
209             if (sortedInternationalStudents == NULL)
210             {
211                 cout << "Your input was invalid. Please re run this program\n";
212                 return -1;
213             }
214
215             cout << "\n\nSorted InternationalStudent Array:\n\n";
216             for (int i = 0; i < numInternationalStudents; i++)
217             {
218                 if (sortedInternationalStudents[i].getFirstName() == "") break;
219                 else cout << sortedInternationalStudents[i];
220             }
221
222             internationalFile.close();
223             delete [] unsortedInternationalStudents;
224             delete [] sortedInternationalStudents;
225         }
226         else
227         {
228             cout << "Your input was invalid. Please re run this program.\n";
229             return -1;
230         }
231
232         return 0;
233     }
234
```