/\*

Michael Dobachesky

SE 124.12

Program 7

PURPOSE:

You have been asked by the Registrar to produce a program which will keep track of the college enrollment activity in a number of courses.

The registrar would like a program which will enable him to enter the name of the class, the instructor's last name, and the number of students enrolled in the class.

Once the data is entered, the registrar would like the ability to produce an alphabetical listing of the classes based on the class name and an alphabetical listing of the classes based on the instructor.

In addition, the registrar would like the ability to search the classes by class name and determine who the instructor is and what the enrollment is in that class.

VARIABLE DICTIONARY:

REPRESENTS TYPE VARIABLE

Is data available? char available\_data

Add another record? char another\_record

Class match for search char class\_match

Menu selection char menu\_choice

Swap data for sort char swap\_data

Number of students enrolled int enrollment\_number

Index number for sort int index\_number

Max enrolled students int max\_enrollment

Number of records processed int records\_processed

Peak for binary search int row\_max

Valley for binary search int row\_min

Subscript\_number int subscript\_number

Total enrolled students int total\_enrollment

Class name string class\_name

Class lookup for Search string class\_search

Duplicate data for sort string duplicate\_data

Instructor name string instructor\_name

\*/

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

char available\_data;

char another\_record;

char class\_match;

char menu\_choice;

char swap\_data;

int enrollment\_number[100];

int index\_number[100];

int max\_enrollment;

int records\_processed;

int row\_max;

int row\_min;

int subscript\_number;

int total\_enrollment;

string class\_name[100];

string class\_search;

string duplicate\_data[100];

string instructor\_name[100];

void setup\_function();

void build\_file();

void load\_file();

void menu\_options();

void classes\_name();

void classes\_instructor();

void sort\_function();

void find\_information();

void successful\_output();

void unsuccessful\_output();

ofstream fout;

ifstream fin;

int main()

{

system("cls");

setup\_function();

cout << "Is there existing college input data available on file? (Y/N): ";

cin >> available\_data;

available\_data = toupper(available\_data);

while (available\_data != 'Y' && available\_data != 'N')

{

cout << "Invalid response " << endl;

cout << "Please enter either a Y or N: ";

cin >> available\_data;

available\_data = toupper(available\_data);

}

if (available\_data == 'N')

{

build\_file();

}

load\_file();

menu\_options();

while (menu\_choice != 'X')

{

switch (menu\_choice)

{

case 'N':

classes\_name();

system("pause");

break;

case 'I':

classes\_instructor();

system("pause");

break;

case 'F':

find\_information();

system("pause");

break;

default:

cout << endl << "Invalid response " << endl;

cout << "Please choose an option from the list (N/I/F/X)" << endl;

system("pause");

break;

}

menu\_options();

}

return 0;

}

void setup\_function()

{

max\_enrollment = 25;

records\_processed = 0;

total\_enrollment = 0;

}

void build\_file()

{

system("cls");

fout.open("school\_data.dat");

subscript\_number = -1;

another\_record = 'Y';

while(another\_record == 'Y')

{

subscript\_number = subscript\_number + 1;

cout << "Class Name: ";

cin >> class\_name[subscript\_number];

cout << "Instructor Name: ";

cin >> instructor\_name[subscript\_number];

cout << "Enrollment: ";

cin >> enrollment\_number[subscript\_number];

while(enrollment\_number[subscript\_number] > max\_enrollment)

{

cout << "Invalid response" << endl;

cout << "Maximum enrollment is limited to " << max\_enrollment << " students" << endl;

cout << "Enrollment: ";

cin >> enrollment\_number[subscript\_number];

}

cout << "Would you like to enter another record? (Y/N): ";

cin >> another\_record;

another\_record = toupper(another\_record);

while (another\_record != 'Y' && another\_record != 'N')

{

cout << "Invalid response" << endl;

cout << "Please enter either a Y or N: ";

cin >> another\_record;

another\_record = toupper(another\_record);

}

system("cls");

fout << class\_name[subscript\_number] << " " << instructor\_name[subscript\_number] << " " << enrollment\_number[subscript\_number] << " " << endl;

}

fout.close();

}

void load\_file()

{

system("cls");

fin.open("school\_data.dat");

subscript\_number = 0;

fin >> class\_name[subscript\_number] >> instructor\_name[subscript\_number] >> enrollment\_number[subscript\_number];

total\_enrollment = total\_enrollment + enrollment\_number[subscript\_number];

while(!fin.eof())

{

subscript\_number = subscript\_number + 1;

fin >> class\_name[subscript\_number] >> instructor\_name[subscript\_number] >> enrollment\_number[subscript\_number];

records\_processed = records\_processed + 1;

total\_enrollment = total\_enrollment + enrollment\_number[subscript\_number];

}

fin.close();

}

void menu\_options()

{

system("cls");

cout << "CLASS MENU" << endl << endl;

cout << "Option" << " " << "Function" << endl << endl;

cout << "N" << " " << "List Classes by Class Name" << endl;

cout << "I" << " " << "List Classes by Instructor" << endl;

cout << "F" << " " << "Find Instructor Name and Enrollment Information" << endl;

cout << "X" << " " << "Exit" << endl << endl;

cout << "Enter Selection: ";

cin >> menu\_choice;

menu\_choice = toupper(menu\_choice);

}

void classes\_name()

{

system("cls");

cout << "ALPHABETICAL CLASS LISTING BY CLASS NAME" << endl << endl;

cout << "Class Name" << " " << "Instructor" << " " << "Enrollment" << endl << endl;

for (subscript\_number = 0; subscript\_number < records\_processed; subscript\_number++)

{

duplicate\_data[subscript\_number] = class\_name[subscript\_number];

}

sort\_function();

for (subscript\_number = 0; subscript\_number < records\_processed; subscript\_number++)

{

cout << class\_name[index\_number[subscript\_number]] << " " << instructor\_name[index\_number[subscript\_number]] << " " << enrollment\_number[index\_number[subscript\_number]] << endl;

}

cout << endl;

cout << "Total Enrollment: " << total\_enrollment << endl << endl;

}

void classes\_instructor()

{

system("cls");

cout << "ALPHABETICAL CLASS LISTING BY INSTRUCTOR NAME" << endl << endl;

cout << "Instructor" << " " << "Class Name" << " " << "Enrollment" << endl << endl;

for (subscript\_number = 0; subscript\_number < records\_processed; subscript\_number++)

{

duplicate\_data[subscript\_number] = instructor\_name[subscript\_number];

}

sort\_function();

for (subscript\_number = 0; subscript\_number < records\_processed; subscript\_number++)

{

cout << instructor\_name[index\_number[subscript\_number]] << " " << class\_name[index\_number[subscript\_number]] << " " << enrollment\_number[index\_number[subscript\_number]] << endl;

}

cout << endl;

cout << "Total Enrollment: " << total\_enrollment << endl << endl;

}

void sort\_function()

{

for (subscript\_number = 0; subscript\_number < records\_processed; subscript\_number++)

{

index\_number[subscript\_number] = subscript\_number;

}

swap\_data = 'Y';

while (swap\_data == 'Y')

{

swap\_data = 'N';

for (subscript\_number = 0; subscript\_number < records\_processed - 1; subscript\_number++)

{

if (duplicate\_data[subscript\_number] > duplicate\_data[subscript\_number + 1])

{

swap (duplicate\_data[subscript\_number], duplicate\_data[subscript\_number + 1]);

swap (index\_number[subscript\_number], index\_number[subscript\_number + 1]);

swap\_data = 'Y';

}

}

}

}

void find\_information()

{

system("cls");

for (subscript\_number = 0; subscript\_number < records\_processed; subscript\_number++)

{

duplicate\_data[subscript\_number] = class\_name[subscript\_number];

}

sort\_function();

cout << "SEARCH FOR INSTRUCTOR AND ENROLLMENT" << endl << endl;

cout << "Please enter a class name to look up: ";

cin >> class\_search;

cout << endl;

class\_match = 'N';

row\_min = 0;

row\_max = records\_processed;

while(class\_match == 'N' && row\_min <= row\_max)

{

subscript\_number = int((row\_min + row\_max) / 2);

if (class\_search < class\_name[index\_number[subscript\_number]])

{

row\_max = subscript\_number - 1;

}

else

{

if (class\_search > class\_name[index\_number[subscript\_number]])

{

row\_min = subscript\_number + 1;

}

else

{

class\_match = 'Y';

}

}

}

if (class\_match == 'Y')

{

successful\_output();

}

else

{

unsuccessful\_output();

}

}

void successful\_output()

{

cout << "The instructor for class " << class\_name[index\_number[subscript\_number]] << " is " << instructor\_name[index\_number[subscript\_number]] << " and there are " << enrollment\_number[index\_number[subscript\_number]] << " students\_enrolled." << endl << endl;

}

void unsuccessful\_output()

{

cout << "The class name you tried to look up was not found " << endl;

cout << "Please try another search " << endl << endl;

}