// Austin Faulkner: May 24, 2020

//------------------------------------------------------------------------------

// QUESTIONS FOR DR. KOH

//------------------------------------------------------------------------------

// 1) How to delete a StudentData object from the LL?

// (REMARK: Function already written in .template file. The names of

// the two methods I wrote for this purpose are

// DeleteFirstTargetNode() and DeleteNode().)

// 2) How do I read a file into the linked list?

// (In particular, how do we read file 'student\_input.txt' into the

// LL?)

// 3) How do we sort StudentData 'student' objects by GPA when StudentData

// attributes are declared PRIVATE?

// (I realize not doing so breaks encapsulation.)

//------------------------------------------------------------------------------

// OBJECTIVE: To simulate a student database system using a linked list complete

// with a sufficient set of LL methods for managing the

// student "database".

//------------------------------------------------------------------------------

//------------------------------------------------------------------------------

// GOALS

//------------------------------------------------------------------------------

//

// 1) LEARN HOW TO DELETE A StudentData "student" OBJECT

// 2) LEARN HOW TO READ DATA FROM A FILE INTO THE StudentData LL

// 3) LEARN HOW TO MAKE StudentData ATTRIBUTES PRIVATE while still sorting.

//

//------------------------------------------------------------------------------

#include <iostream>

#include <string>

#include <fstream>

#include <iomanip> // setw()

#include <cctype> // toupper

#include <cstdlib> // EXIT\_SUCCESS

#include "Templated\_LL.h"

#include "StudentData.h"

using namespace My\_Templated\_LL;

void displayMenu();

int main(int argc, char\* argv[])

{

LinkedList<StudentData> list;

ListNode<StudentData> \*pointer = nullptr;

StudentData student;

std::ofstream fout;

std::ifstream fin;

char choice = ' ';

double minGPA = 0.00,

maxGPA = 0.00,

dbl = 0.00;

int message = 0;

do

{

displayMenu();

std::cin >> choice;

std::cout << std::endl;

switch (choice)

{

case 'a':

case 'A': std::cout << "The choice you entered is: "

<< choice << std::endl << std::endl

<< "There are " << list.FindListLength(pointer)

<< " students enrolled at Texas State University.";

break;

case 'b':

case 'B': std::cout << "The choice you entered is: "

<< choice << std::endl

<< "Answer: " << list.IsSortedUp(pointer)

<< std::endl;

break;

case 'c':

case 'C': std::cout << "The choice you entered is: "

<< choice << std::endl << std::endl;

std::cout << "Please enter student data "

<< std::endl

<< "in the following format: "

<< std::endl << std::endl

<< "Name:"

<< std::endl

<< "Student ID: "

<< std::endl

<< "Major (Initial(s)): "

<< std::endl

<< "GPA: "

<< std::endl

<< "Residency Status (State): "

<< std::endl

<< "Street Address: "

<< std::endl

<< "P.O. Box / Apt. #: "

<< std::endl

<< "City State Zipcode: "

<< std::endl << std::endl;

std::cin >> student;

list.InsertAsHead(pointer, student);

break;

case 'd':

case 'D': std::cout << "The choice you entered is: "

<< choice << std::endl << std::endl;

std::cout << "Please enter student data "

<< std::endl

<< "in the following format: "

<< std::endl << std::endl

<< "Name:"

<< std::endl

<< "Student ID: "

<< std::endl

<< "Major (Initial(s)): "

<< std::endl

<< "GPA: "

<< std::endl

<< "Residency Status (State): "

<< std::endl

<< "Street Address: "

<< std::endl

<< "P.O. Box / Apt. #: "

<< std::endl

<< "City State Zipcode: "

<< std::endl << std::endl;

std::cin >> student;

list.InsertAsTail(pointer, student);

break;

case 'e':

case 'E': std::cout << "The choice you entered is: "

<< choice << std::endl << std::endl;

std::cout << "Please enter the student data you "

<< "wish to insert in the following format: "

<< std::endl << std::endl

<< "Name:"

<< std::endl

<< "Student ID: "

<< std::endl

<< "Major (Initial(s)): "

<< std::endl

<< "GPA (Enter 0.00): "

<< std::endl

<< "Residency Status (State): "

<< std::endl

<< "Street Address: "

<< std::endl

<< "P.O. Box / Apt. #: "

<< std::endl

<< "City State Zipcode: "

<< std::endl << std::endl;

std::cin >> student;

std::cout << std::endl

<< "Enter the student's GPA: ";

std::cin >> dbl;

list.InsertSortedUp(pointer, student, dbl);

break;

//-----------------------------------------------------------------------------

// Delete Method, DeleteFirstTargetNode(), Below

//-----------------------------------------------------------------------------

case 'f':

case 'F': std::cout << "The choice you entered is: "

<< choice << std::endl << std::endl;

std::cout << "Enter the student you wish to unenroll: "

<< std::endl << std::endl

<< "Name:"

<< std::endl

<< "Student ID: "

<< std::endl

<< "Major (Initials): "

<< std::endl

<< "GPA: "

<< std::endl

<< "Residency Status (State): "

<< std::endl

<< "Street Address: "

<< std::endl

<< "P.O. Box / Apt. #: "

<< std::endl

<< "City State Zipcode: "

<< std::endl << std::endl;

std::cin >> student;

list.DeleteFirstTargetNode(pointer, student);

// Needs to be fixed

break;

//------------------------------------------------------------------------------

case 'g':

case 'G': std::cout << "The choice you entered is: "

<< choice << std::endl << std::endl

<< "Here is all student data: " << " ";

list.ShowAll(std::cout, pointer);

break;

case 'h':

case 'H': std::cout << "The choice you entered is: "

<< choice << std::endl;

list.FindMinMax(pointer, minGPA, maxGPA);

break;

case 'i':

case 'I': std::cout << "The choice you entered is: "

<< choice << std::endl

<< "The average gpa at Texas State University is: ";

std::cout << list.FindAverage(pointer);

break;

case 'j':

case 'J': std::cout << "The choice you entered is: "

<< choice << std::endl

<< "The student records are now sorted."

<< std::endl;

list.SortLinkedList(pointer);

break;

case 'k':

case 'K': std::cout << "The choice you entered is: "

<< choice << std::endl << std::endl;

std::cout << "Please enter existing student data "

<< std::endl

<< "in the following format: "

<< std::endl << std::endl

<< "Name:"

<< std::endl

<< "Student ID: "

<< std::endl

<< "Major (Initial(s)): "

<< std::endl

<< "GPA: "

<< std::endl

<< "Residency Status (State): "

<< std::endl

<< "Street Address: "

<< std::endl

<< "P.O. Box / Apt. #: "

<< std::endl

<< "City State Zipcode: "

<< std::endl << std::endl;

std::cin >> student;

list.PromoteTarget(pointer, student);

// IS promoting existing members of LL roster;

// IS NOT promoting non-members of LL roster.

break;

case 'l':

case 'L': std::cout << "The choice you entered is: "

<< choice << std::endl;

list.ListClear(pointer, message);

break;

//------------------------------------------------------------------------------

// File-Read-Into-LL Method, ReadStudentRecords(), Below

//------------------------------------------------------------------------------

case 'm':

case 'M': std::cout << "The choice you entered is: "

<< choice << std::endl << std::endl;

list.ReadStudentRecords(fin, std::cin, student);

// Need to put data into LL roster somehow. Not sure how.

break;

//------------------------------------------------------------------------------

case 'n':

case 'N': std::cout << "The choice you entered is: "

<< choice << std::endl << std::endl

<< "All student records printed to file "

<< "\'student\_output.txt.\'";

list.PrintStudentRecords(fout, pointer);

break;

}

} while (toupper(choice) != 'Q');

return EXIT\_SUCCESS;

}

void displayMenu()

{

std::cout << std::endl << std::endl

<< std::setw(61) << "Texas State University" << std::endl

<< std::setw(50) << "---------------------------------------"

<< "---------------------------------------"

<< std::endl;

std::cout << std::setw(79)

<< "A) Find the number of students at Texas State University."

<< std::endl

<< std::setw(63)

<< "B) Check if the student roster is sorted." << std::endl

<< std::setw(71)

<< "C) Insert student at the beginning of the roster."

<< std::endl

<< std::setw(65)

<< "D) Insert student at the end of the roster." << std::endl

<< std::setw(80)

<< "E) Insert a student in the sorted bank of student records."

<< std::endl

<< std::setw(42)

<< "F) Delete a student." << std::endl

// NEED HELP WITH DELETING A STUDENT - .template function is DONE

<< std::setw(47)

<< "G) Show all student data." << std::endl

<< std::setw(59)

<< "H) Find the minimum and maximum GPAs." << std::endl

<< std::setw(76)

<< "I) Find the average of the GPAs in the student roster."

<< std::endl

<< std::setw(57)

<< "J) Sort the student records by GPA." << std::endl

// COULD WE MAKE THE ATTRIBUTES IN StudentData.h private SOMEHOW?

<< std::setw(80)

<< "K) Promote target student by GPA to the front of the list."

// ALMOST DONE; JUST NEED TO DEAL WITH CASE WHEN DATA IS NOT IN LL.

<< std::endl

<< std::setw(53)

<< "L) Clear the whole linked list." << std::endl

<< std::setw(58)

<< "M) Read student records from a file." << std::endl

// NEEDS WORK INPUTTING DATA INTO LL

<< std::setw(57)

<< "N) Print student records to a file." << std::endl

<< std::setw(42)

<< "Q) Quit the program."

<< std::endl;

std::cout << std::setw(50) << "---------------------------------------"

<< "---------------------------------------"

<< std::endl << std::endl;

}