Edwin Jones

CS300 3-3: Project One Milestone One

//Defaults to public members

struct CourseInfo {

string code

string name

//accounts for 0-N prerequisites

vector<string> prerequisites

void printSortedVector(vector<CourseInfo> &courses) {

sort using built in sort function

for each course in courses

print course code

print couse name

for each prereq in course prerequisites

if prereq length is 0

continue

if prereq is the last in the vector

print prereq

else

print prereq with a following comma

}

//Read CSV file then add row to vector of struct courseInfo

void loadCSVToVector(string csvFile, vector<CourseInfo> &courses) {

ifstream file(csvFile)

string line

if file is not open

throw an error

while there is a row, assign the content of the row to line {

if the length of the line is 0

continue

if the end of the line doesn’t have a comma

add a comma to the end

if the number of commas found is < 2

throw an error

initialize CourseInfo instance course

set course.code = string before first comma

delete index 0 to index of comma

set couse.name = string before first comma

delete index 0 to index of comma

while there are more than 1 chars in line

if there is no comma in line

add line to end of course.prerequisites vector

break

add string before next comma to the end of the course.prerequisites vector

set line = line with index 0 index of comma deleted

add course to end of courses vector

close file

}

// Verify that each unique course prerequisite is listed as a course

void validateCourses(vector<CourseInfo> &courses){

vector<string> tempPrereqs

vector<string> tempCourses

//courses listed to check prerequisite courses are listed

for each CourseInfo course instance in the courses instance

add course.code to the end of the tempCourses vector

//check each course for their prerequisites

for each CourseInfo course instance in the courses instance

//check if prerequisite course is listed as course

for each string prereq in course.prerequisites

if prereq is in tempCourses

continue

else

throw an error

}

// Search for course given course code as search parameter

void searchCourse(vector<CourseInfo> people, string searchItem)

for each CourseInfo course instance in courses

if each course.code = searchItem

print course.code

print course.name

for each string prereq in course.prerequisites

if prereq length is 0

continue

print prereq

return

print “Course code not found”

main {

set string csvFile = csv file location

create vector<CourseInfo> courses

set userChoice as 0

while userChoice != 9

print menu

set userChoice to user input

switch(userChoice)

case 1: // Load the file data into the data structure

call the loadCSVToVector(csvFile, courses) method

case 2:

call the validateCourses(courses) method

case 3: // Print an alphanumerically ordered list

call the printSortedVector(courses) method

case 4: // Print the course title and the prerequisites for any individual course

call the searchCourse(people, get user search parameter) method

return 0

}