CS 300 Project One Milestone One

Pseudocode

// Define the structure to hold course information

struct Course

string courseNumber

string courseName

vector<string> prerequisites

// Function to read and parse the file

function readFile(fileName: string) -> vector<Course>

vector<Course> courses

open file fileName

if file is not open

print "Error: Unable to open file"

return courses

while not end of file

line = read line from file

courseData = split line by comma

if length of courseData < 2

print "Error: Invalid line format"

continue

course = Course()

course.courseNumber = courseData[0]

course.courseName = courseData[1]

for i from 2 to length of courseData - 1

course.prerequisites.push\_back(courseData[i])

courses.push\_back(course)

close file

return courses

// Function to validate the courses

function validateCourses(courses: vector<Course>) -> bool

courseNumbers = create empty set

for course in courses

courseNumbers.add(course.courseNumber)

for course in courses

for prerequisite in course.prerequisites

if prerequisite not in courseNumbers

print "Error: Prerequisite", prerequisite, "for course", course.courseNumber, "does not exist"

return false

return true

// Function to search for a course and print its information

function searchCourse(courses: vector<Course>, courseNumber: string)

for course in courses

if course.courseNumber == courseNumber

print "Course Number:", course.courseNumber

print "Course Name:", course.courseName

print "Prerequisites:"

for prerequisite in course.prerequisites

print " ", prerequisite

return

print "Course", courseNumber, "not found"

// Main function

function main()

fileName = "courses.txt"

courses = readFile(fileName)

if not validateCourses(courses)

print "Course file validation failed"

return

searchCourse(courses, "CSCI300")

searchCourse(courses, "MATH201”)

Explanation

To help ABC University advisors access course information, we need to create a program that reads data from a file and stores it in a vector data structure. This program will also check if the data is correct and allow users to find specific course information.

First, we need to define a structure called Course. This structure will hold information about each course, including the course number, course name, and a list of prerequisites (courses that must be taken before the course). Next, we need a function to read the file containing the course information. This function will open the file, read each line, and split the line by commas to separate the course number, course name, and prerequisites. If a line doesn't have at least two items (course number and course name), it will be ignored. For each valid line, a Course object will be created and added to a list of courses.

After reading the file, we need to make sure the data is correct. We'll create a function that checks if all prerequisites listed for each course actually exist in the list of courses. If any prerequisite is missing, an error message will be shown, and the validation will fail. We also need a function to search for a specific course by its course number. This function will go through the list of courses, and if it finds the course, it will print the course number, name, and its prerequisites. If the course isn't found, it will print a message saying the course wasn't found.

Finally, in the main part of the program, we'll specify the file name, read the file to get the course data, validate the courses, and then search for some specific courses to print their information. This approach ensures that the program can read, validate, and search course information effectively, making it easier for advisors to access the data they need.