Main Function:

1. Prompt the user for the file path (or use a default path if none is provided).
2. Call openFile() with the file path.
3. If the file opens successfully:
   * Call parseFile() to read and process each line.
   * Call validateFile() to check formatting and prerequisites.
4. Prompt the user for a course to search.
5. Call printCourse() with the user’s search term.

openFile(filePath: String):

1. Try to open the file at the given path.
2. If the file cannot be opened:
   * Print an error message.
   * Exit the program.
3. If the file opens successfully, return the file handle.

parseFile(fileHandle):

1. Create an empty courseList (a vector of Course objects).
2. Loop through each line in the file:
   * Split the line by commas (or another delimiter) into tokens.
   * Ensure there are at least 2 tokens (course number and title).
   * Create a new Course object:
     + Assign the first token to courseID.
     + Assign the second token to courseName.
     + If additional tokens exist:
       - For each additional token, append it to the preList.
     + Set preCount to the number of prerequisites.
   * Add the Course object to courseList.
3. Return courseList.

validateFile(courseList):

1. Set valid to True.
2. Loop through each course in courseList:
   * If courseID or courseName is missing, set valid to False and print an error.
   * Loop through each course's preList:
     + Call searchList(preCourseID) to find the prerequisite.
     + If the prerequisite is not found, set valid to False and print an error.
3. If valid is False, print "File validation failed" and exit the program.
4. If valid is True, print "File validated successfully."

Course Object Definition:

plaintext

Copy code

struct Course {

courseID

courseName

preCount

preList

Course() {

courseID = ""

courseName = ""

preCount = 0

preList = []

}

}

searchList(searchTerm: String):

1. Loop through courseList:
   * If courseID matches searchTerm, return the corresponding Course object.
2. If no match is found, return null.

printCourse(searchTerm: String):

1. Call searchList(searchTerm) to retrieve the course.
2. If the course is found:
   * Print courseID, courseName, and preList.
   * If the course has prerequisites:
     + For each prerequisite in preList, recursively call printCourse(preCourseID).
3. If the course is not found, print "Course not found."

Notes:

* The openFile() function handles file reading and basic error checking for file access.
* The parseFile() function processes each line, creates Course objects, and stores them in a vector.
* The validateFile() function ensures that all lines are properly formatted and that all prerequisites exist as courses in the file.
* The searchList() function is used for finding a course in the list by its ID.
* The printCourse() function prints course details and prerequisites recursively, if applicable.