Victor Udeh  
CS300 M3-3  
JAN 28, 2024  
Vector Data Structure Pseudocode

PSEUDOCODE  
  
// Define a Course structure with ID, name, prerequisite count, and a list of prerequisites

struct Course {

courseID: String

courseName: String

preCount: Integer

preList: List<String>

// Constructor initializes a Course with default values

Constructor() {

courseID = ""

courseName = ""

preCount = 0

preList = new List<String>()

}

}

// Main function of the program

function main() {

// Create a list to store Course objects

courseList = new List<Course>()

// Get the CSV file path from the user or use a default path

filePath = getUserInputForCSVFilePath()

if filePath is empty

filePath = "default/path/to/csv"

// Parse the CSV file and validate the course list

courseList = txtParser(filePath)

if not validateList(courseList)

print "Invalid course list. Please check the CSV file."

return

// Get course ID from user and print course information

userSearch = getUserInputForCourseID()

printCourse(userSearch, courseList)

}

// Function to parse the CSV file and return a list of Course objects

function txtParser(filePath: String) -> List<Course> {

tempList = new List<Course>()

file = open CSV file at filePath

while not end of file

row = read next row from file

if row has at least 2 non-empty strings

course = new Course()

course.courseID = row[0]

course.courseName = row[1]

for each additional column in row

if column is not empty

course.preCount++

course.preList.add(column)

tempList.add(course)

close file

return tempList

}

// Function to search for a course by ID in the course list

function searchList(courseID: String, courseList: List<Course>) -> Course {

for each course in courseList

if course.courseID equals courseID

return course

return new Course() // Return an empty course if not found

}

// Function to print information about a course

function printCourse(courseID: String, courseList: List<Course>) {

course = searchList(courseID, courseList)

if course.courseID is not empty

print course.courseID, course.courseName

for each preID in course.preList

printCourse(preID, courseList)

}

// Function to validate the course list

function validateList(courseList: List<Course>) -> Boolean {

for each course in courseList

for each preID in course.preList

if searchList(preID, courseList).courseID is empty

return False

return True

}