

Leslie Jaquez

CS-300: Analysis and Design

Southern New Hampshire University

Sunday, October 13, 2024

## 6-2 Project One Pseudocode

Function readDataFromFile(filename):

File = open(filename, "r")

If file is not null:

For line in file:

If isValidFormat(line):

Course = createCourseObject(line)

addCourseToDataStructure(course)

Else:

print("Formatting error in line:",line)

Else:

print("Error opening file:" filename)

close(file)

Function isValidFormat(line):

//Check if the line has the correct format

//Return true if valid, false otherwise

Function createCourseObject(line):

//Parse the line and create a Course object with the data

//Return the created Course Object

Function addCourseToDataStructure(course):

    //Add the course to the appropriate data structure

Function printCourse(course):

    print("Course Title:", course.getTitle())

    print("Prerequisites:", course.getPrerequisites())

Function printCourseList(dataStructure):

    sortedCourses = sortCourses(dataStructure)

    For course in sortedCourses:

        printCourse(course)

Function displayMenu():

    While True:

        Print ("Menu:")

        Print ("1. Load Data Structure")

        Print ("2. Print Course List")

        Print ("3. Print Course")

        Print ("4. Exit")

Function getMenuChoice():

    Choice = -1

    While choice < 1 or choice >4:

        displayMenu()

        Choice = getUserInput ("Enter your choice:")

    Return choice

Function processMenuChoice(choice):

    If choice ==1:

        Filename = getUserInput("Enter the filename: ")

        readDataFromFile(filename)

    Else if choice ==2:

        printCourseList(dataStructure)

    Else if choice ==3:

        courseCode = getUserInput("enter the course code: ")

        Course = findCourse(dataStructure, courseCode)

        If course is not null:

            printCourse(course)

        Else:

            print("Course not found.")

    Else if choice == 4:

        exit Program()

Function main():

    Choice = -1

    While choice != 4:

        Choice = getMenuChoice()

        processMenuChoice(choice)

Function sortCourses(dataStructure):

    //sort the courses in the data structure by alphanumeric course number

    //return the sorted list of courses

Function printSortedCourseList(dataStructure):

    sortedCourses = sortCourses(dataStructure)

    For course in sortedCourses:

        printCourse(course)

Runtime evaluation:

Assuming that there are  $n$  courses stored within the data structure, the best way to examine the runtime is to evaluate the files and constructing course objects. This can include examining file formats for errors, and additional course object creation and data structure additions. Both of these steps could add an additional dollar per step. Each data structure has its advantages and disadvantages. Vectors provide sorting support, enables quick access to elements by index, and enables order for insertion. However, it does not allow for multiline searching as it is done in a straight line. Hashing tables allow quick access and element recovery while being able to manage massive columns of data. A con for them is that elements are not kept in any order and may impact performance but collision resolution. Binary Search Trees offer quick insertion, allowing for ordered traversal and user friendly sortability. However, its biggest drawback is the requirement of additional RAM to withhold node links and maintain peak performance in order to be balanced.