Joshua Hale

July 11, 2024

**CS300**

**4-3 Project One Milestone Two: Hash Table Data Structure**

A diagram of a blockchain

Description automatically generated

**Project One Milestone Two: Pseudocode for ABC University**

**Overview**

This document provides pseudocode for a program designed to help ABC University's Computer Science advisors access and manage course information for students. The pseudocode demonstrates how to load data from a file into a hash table, create course objects, and print course information along with prerequisites.

***Pseudocode Document for Milestone Two***

**1. Open File and Read Data**

The first step is to open the file containing the course information and read each line from the file.

1. Open the file containing course information.

2. Read each line from the file.

**2. Parse and Validate Data**

Next, we need to parse each line to extract the course number, course name, and any prerequisites. We also need to validate the data to ensure it is correctly formatted and that any prerequisites listed exist as courses in the file.

3. For each line:

a. Split the line by commas to separate the course number, course name, and prerequisites.

b. Ensure that there are at least two pieces of information (course number and course name).

c. Validate that any listed prerequisite exists as a course in the file.

- Store the course number and its details in a temporary list for quick lookup of prerequisites.

d. If validation fails, print an error message and skip the line.

**3. Create Course Objects and Store in Hash Table**

Once the data is validated, we create course objects and store them in a hash table. Each course object will hold the data from a single line of the input file.

4. Define a `Course` structure with fields for course number, course name, and prerequisites.

5. For each valid line:

a. Create a `Course` object.

b. Store the course number, name, and prerequisites in the `Course` object.

c. Insert the `Course` object into the hash table using the course number as the key.

**4. Print Course Information**

Finally, we define a function to search for and print course information from the hash table, including any prerequisites.

6. Define a function to search for and print course information:

a. Input: Course number.

b. Retrieve the `Course` object from the hash table using the course number.

c. Print the course number, name, and prerequisites.

d. For each prerequisite, recursively print its course information.

**Detailed Pseudocode**

Here is the detailed pseudocode for the entire process:

// Function: Load Data into Hash Table

function loadData(filePath):

// Step 1: Open the file

file = open(filePath)

courseList = createEmptyList()

hashTable = createHashTable()

// Step 2: Read each line from the file

for line in file:

// Split the line by commas

data = split(line, ',')

// Step 3: Validate the line

if length(data) < 2:

print("Error: Invalid line format.")

continue

courseNumber = data[0]

courseName = data[1]

prerequisites = data[2:] // Remaining elements are prerequisites

// Check if all prerequisites exist in the file

for prereq in prerequisites:

if not prereq in courseList:

print("Error: Prerequisite", prereq, "does not exist.")

continue

// Add course to the list for validation

addToList(courseList, courseNumber)

// Step 4: Create a Course object

course = Course(courseNumber, courseName, prerequisites)

// Insert the course into the hash table

hashTable[courseNumber] = course

// Close the file

close(file)

return hashTable

// Function: Print Course Information

function printCourseInfo(hashTable, courseNumber):

// Step 5: Retrieve the course from the hash table

course = hashTable[courseNumber]

if course is None:

print("Course not found.")

return

// Print course information

print("Course Number:", course.courseNumber)

print("Course Name:", course.courseName)

print("Prerequisites:")

for prereq in course.prerequisites:

print(" -", prereq)

// Step 6: Recursively print prerequisite info

printCourseInfo(hashTable, prereq)

// Step 4: Course Structure

class Course:

courseNumber

courseName

prerequisites

function Course(courseNumber, courseName, prerequisites):

this.courseNumber = courseNumber

this.courseName = courseName

this.prerequisites = prerequisites

**Explanation**

1. **Loading Data**:

The pseudocode begins by opening a file and reading each line. It then splits each line by commas to separate the course number, course name, and prerequisites. The data is validated to ensure it is correctly formatted and that any prerequisites exist as courses in the file. If the validation fails, an error message is printed, and the line is skipped.

1. **Creating Course Objects**:

After validation, a Course object is created for each valid line. This object holds the course number, course name, and prerequisites. Each course object is then inserted into a hash table using the course number as the key.

1. **Printing Course Information**:

A function is defined to search for and print course information from the hash table. The function takes a course number as input, retrieves the corresponding course object from the hash table, and prints the course number, name, and prerequisites. For each prerequisite, the function recursively prints its information.

***Image Reference:***

<https://khalilstemmler.com/blogs/data-structures-algorithms/hash-tables/>