Joshua Hale

July 11, 2024

**CS300**

**3-3 Project One Milestone One: Vector Data Structure**



**Pseudocode**

**1. Opening the File and Reading Data**

BEGIN

OPEN file "course\_data.txt"

IF file is not found THEN

PRINT "Error: File not found"

RETURN

END IF

CREATE vector<Course> courses

WHILE (not end of file)

READ line from file

SPLIT line by comma into tokens

IF (number of tokens < 2) THEN

PRINT "Error: Invalid line format"

CONTINUE

END IF

SET courseNumber = tokens[0]

SET courseName = tokens[1]

CREATE vector<String> prerequisites

FOR i = 2 to (number of tokens - 1)

ADD tokens[i] to prerequisites

END FOR

CREATE Course object with courseNumber, courseName, prerequisites

ADD Course object to courses vector

END WHILE

CLOSE file

END

**2. Validating the File Format**

BEGIN

FUNCTION validateFile(Vector<Course> courses)

CREATE set<String> courseNumbers

FOR each course in courses

ADD course.courseNumber to courseNumbers

END FOR

FOR each course in courses

FOR each prerequisite in course.prerequisites

IF prerequisite NOT IN courseNumbers THEN

PRINT "Error: Prerequisite " + prerequisite + " does not exist"

RETURN FALSE

END IF

END FOR

END FOR

RETURN TRUE

END FUNCTION

END

**3. Creating Course Objects and Storing Them**

BEGIN

FUNCTION createCourse(String courseNumber, String courseName, Vector<String> prerequisites)

CREATE Course object

SET object.courseNumber = courseNumber

SET object.courseName = courseName

SET object.prerequisites = prerequisites

RETURN object

END FUNCTION

FOR each line in file

SPLIT line by comma into tokens

SET courseNumber = tokens[0]

SET courseName = tokens[1]

CREATE vector<String> prerequisites

FOR i = 2 to (number of tokens - 1)

ADD tokens[i] to prerequisites

END FOR

CREATE course object using createCourse function

ADD course object to courses vector

END FOR

END

**4. Searching for a Specific Course and Printing Information**

BEGIN

FUNCTION searchCourse(Vector<Course> courses, String courseNumber)

FOR each course in courses

IF course.courseNumber == courseNumber THEN

PRINT "Course Number: " + course.courseNumber

PRINT "Course Name: " + course.courseName

PRINT "Prerequisites: "

IF course.prerequisites.size() == 0 THEN

PRINT "None"

ELSE

FOR each prerequisite in course.prerequisites

PRINT prerequisite

END FOR

END IF

RETURN

END IF

END FOR

PRINT "Course not found"

END FUNCTION

END

**Pseudocode for Non-Technical Professionals**

1. **Opening the File and Reading Data**
   * Start by opening a file named "course\_data.txt".
   * If the file can't be found, show an error message saying "Error: File not found" and stop.
   * Create a list to store course information.
   * While there is more data to read in the file:
     + Read a line from the file.
     + Split the line into parts using commas.
     + If there are less than 2 parts, show an error message saying "Error: Invalid line format" and continue to the next line.
     + Extract the course number from the first part and the course name from the second part.
     + Create a list to store prerequisites.
     + For each part after the second one, add it to the prerequisites list.
     + Create a course object with the course number, course name, and prerequisites.
     + Add this course object to the list of courses.
   * Close the file.
2. **Validating the File Format**
   * Define a function to validate the course data:
     + Create a set to store all course numbers.
     + For each course in the list of courses:
       - Add the course number to the set.
     + For each course in the list of courses:
       - For each prerequisite of the course:
         * If the prerequisite is not in the set of course numbers, show an error message saying "Error: Prerequisite [prerequisite] does not exist" and return false.
     + Return true if all prerequisites exist.
3. **Creating Course Objects and Storing Them**
   * Define a function to create a course object:
     + Create a course object.
     + Set the course number, course name, and prerequisites for the course object.
     + Return the course object.
   * For each line in the file:
     + Split the line into parts using commas.
     + Extract the course number from the first part and the course name from the second part.
     + Create a list to store prerequisites.
     + For each part after the second one, add it to the prerequisites list.
     + Create a course object using the function to create course objects.
     + Add this course object to the list of courses.
4. **Searching for a Specific Course and Printing Information**
   * Define a function to search for a specific course:
     + For each course in the list of courses:
       - If the course number matches the given course number:
         * Print the course number, course name, and prerequisites.
         * If there are no prerequisites, print "None".
         * Otherwise, print each prerequisite.
         * Stop searching.
     + If the course is not found, print "Course not found".

This pseudocode is simplified to make it easier for non-technical professionals to understand the overall process and logic without going into technical details.

Image Reference:

<https://linuxhint.com/sort-cpp-vectors/>