5-3 Project One Milestone Three

Jorgo Qendro

Southern New Hampshire University

CS-300: Analysis and Design

Professor Adamo

October 3rd,2024

Project Milestone Pseudocpde

function loadCourses(file):

open file

for each line in file:

split line by commas into courseData

if length of courseData < 2:

print "Error: Incorrect format, each course must have a course number and title"

else:

courseNumber = courseData[0]

courseTitle = courseData[1]

prerequisites = courseData[2:] # Get all prerequisites

if prerequisites are not empty:

for each prerequisite in prerequisites:

if prerequisite not in courses:

print "Error: Prerequisite", prerequisite, "is not a valid course"

create course object with courseNumber, courseTitle, and prerequisites

insert course object into binary search tree (BST)

function validateCourseData(courseData):

if length of courseData < 2:

return False # Not enough parameters

courseNumber = courseData[0]

prerequisites = courseData[2:]

if prerequisites are not empty:

for each prerequisite in prerequisites:

if prerequisite not in file:

return False # Invalid prerequisite

return True

function createCourse(courseData):

courseNumber = courseData[0]

courseTitle = courseData[1]

prerequisites = courseData[2:]

newCourse = Course(courseNumber, courseTitle, prerequisites)

return newCourse

function insertCourseIntoTree(courseTree, course):

if tree is empty:

set root to course

else:

recursively insert course based on courseNumber

function searchCourse(Tree<Course> courses, String courseNumber):

find the course in the BST by courseNumber

if course is found:

print course number and title

if course has prerequisites:

for each prerequisite:

searchCourse(courses, prerequisite)

else:

print "Course not found"

function printCourseInfo(course):

print "Course Number:", course.courseNumber

print "Course Title:", course.courseTitle

if course has prerequisites:

print "Prerequisites:"

for each prerequisite in course.prerequisites:

print prerequisite

else:

print "No prerequisites"

### Additional Notes:

* **Error Handling:** The pseudocode includes basic validation for course formatting and prerequisites.
* **Data Structure:** The course objects are stored and managed using a binary search tree (BST).
* **Time Complexity:** The time complexity for both searching and displaying the course data is O(n), where *n* represents the number of courses. Each course is inserted and searched in the tree exactly once.