Carlos Rodriguez

Professor Yurik

CS 300

November 26, 2024

Project One Milestone Three

void searchCourse(Tree<Course> courses, String courseNumber) {

**Part One: Open, Read, Parse and Check the File Formats**

Function that will load the course data from the file Path:

// Opens the file

Try to open the file at the file Path

If the file fails to open,

Print an error message for the file

Exit out of the function

// Create the empty binary search tree (BST) for the program

Initialize the BST Root equal to null

// Run through each line in the file

For each line in the data,

Split the line into tokens by commas by setting tokens equal to the line split

// Validate the correct line format of the file

If the length of tokens is less than two:

Print an invalid format message

Keep checking on the next line

// Extract the course data

Set the course number equal to the token’s placement on line 1 [0]

Set the course name equal to the token’s placement on line 2 [1]

Set the prerequisites equal to the token’s placement on line 3 and after [2:]

// Validate the prerequisites for the courses

If prerequisite is not found in the tree:

Print an error message and state that this course does not need one.

Move to the next data set

// If validation passes, make the new course and add the course to the data

Create a Course object with the course number, course name, and the prerequisites

Set the BST Root equal to the Insert course into the root and course data

Close the file and return the BST

**Part Two: Create Course Objects and Store into Data Structure**

Structure the Course Information:

Create a string of the course number

Create a string of the course name

Create a list of the prerequisites

Structure the tree node for the BST:

Create the course object

Set the tree node pointer to the left child

Set the tree node pointer to the right child

Make a function that will create the course with the course number, course name & prerequisites

Initialize the course object

Set the course object of course number equal to the course number

Set the course object of course name equal to the course name

Set the course object of course prerequisites equal to the course prerequisites

Return the course object

Make a function that will insert the course into a node of the course data

If node is equal to null:

Create a new tree node with course and set left and right to null

Return the new node

If the course number is less than the node of the course number:

Add the node to the left

Else if the course number is greater than the node of the course number:

Add the node to the right

Return node

**Part Three: Print out Course Information and Prerequisites, if applicable**

Create a function to print out all the listed courses(node):

If the node is equal to null:

Return

// Perform in-order traversal to print courses in sorted order

Print all the course starting with the left child

Print the course number

Print the course title

If prerequisites are not empty:

Print all the prerequisites of the course

Else:

Print that there is no prerequisites

Print all the course of the right child

Return the courses