# CS 300 Pseudocode Document

FUNCTION searchCourse(BST<Course> courseTree, String courseNumber)

courseFound = courseTree.find(courseNumber) // Use BST search

IF courseFound IS NOT NULL

PRINT "Course Information:"

PRINT courseFound.number + " - " + courseFound.title

IF NOT courseFound.prerequisites.empty()

PRINT "Prerequisites:"

FOR EACH prereq IN courseFound.prerequisites

PRINT prereq

ENDFOR

ENDIF

ELSE

PRINT "Error: Course not found."

ENDIF

ENDFUNCTION

// Main Program

BEGIN

fileName = "course\_information.txt" // File to be processed

courseTree = new BST<Course>() // Create an empty binary search tree

// File Input and Validation (Similar to previous, but storing in BST)

fileHandle = OPEN(fileName)

IF fileHandle IS NULL

PRINT "Error opening file: " + fileName

RETURN NULL

ENDIF

WHILE NOT END\_OF\_FILE(fileHandle)

line = READ\_LINE(fileHandle)

// ... (Error checks for empty lines and insufficient parameters)

courseNumber = courseData[0]

courseTitle = courseData[1]

newCourse = new Course(courseNumber, courseTitle)

// Handle Prerequisites with Validation

FOR i = 2 TO courseData.length - 1

prerequisite = courseData[i]

// Defer prerequisite validation until after all courses are loaded

newCourse.addPrerequisite(prerequisite) // Add even if not found yet

ENDFOR

courseTree.insert(newCourse) // Insert into the BST

ENDWHILE

CLOSE(fileHandle)

// Validate Prerequisites After File Reading

// (Same logic as in Milestone 2, but using courseTree.find() instead of containsKey())

FOR EACH course IN courseTree // Use in-order traversal to get sorted order

FOR EACH prerequisite IN course.prerequisites

IF courseTree.find(prerequisite) IS NULL // Check if prerequisite exists in the tree

PRINT "Error: Prerequisite " + prerequisite + " for course " + course.number + " does not exist."

// Optional: Handle error (remove invalid prerequisite, etc.)

ENDIF

ENDFOR

ENDFOR

// Search and Print Course Information (BST Version)

PRINT "Enter course number to search:"

courseToSearch = READ\_INPUT()

searchCourse(courseTree, courseToSearch)

END