# CS-300 Milestone Three: Tree Data Structure Pseudocode

This milestone outlines how the advising program for ABC University loads course data, creates course objects, and stores them in a binary search tree (BST). The pseudocode below is based on the same logic I used later in the final project. It shows how data is read, validated, inserted, and displayed using the BST structure. Each section directly answers one of the assignment prompts.

## 1. Open, Read, and Validate File Data

function loadCoursesFromFile(filename):  
 open the file  
 if file cannot be opened:  
 print "error opening file"  
 return  
  
 make an empty list called allCourses  
  
 for each line in file:  
 split line by commas into fields  
 if fields.size < 2:  
 print "bad line format"  
 continue  
  
 courseNumber = fields[0]  
 courseName = fields[1]  
 prereqs = rest of fields (if any)  
  
 make new Course  
 set courseNumber, name, prerequisites  
 add to allCourses  
  
 // check that prerequisites exist  
 for each course in allCourses:  
 for each prereq in course.prerequisites:  
 if prereq not in allCourses.courseNumber:  
 print "missing prerequisite: " + prereq  
  
 return allCourses

## 2. Create Course Objects and Store Them in the Tree

struct Course {  
 string courseNumber  
 string name  
 list<string> prerequisites  
}  
  
function buildCourseBST(allCourses):  
 make an empty BST called root  
 for each course in allCourses:  
 root = insert(root, course)  
 return root  
  
function insert(node, course):  
 if node == null:  
 return new Node(course)  
 if course.courseNumber < node.data.courseNumber:  
 node.left = insert(node.left, course)  
 else:  
 node.right = insert(node.right, course)  
 return node

## 3. Print Course Information and Prerequisites

function searchCourse(root, courseNumber):  
 node = root  
 while node != null:  
 if courseNumber == node.data.courseNumber:  
 printCourse(node.data)  
 return  
 else if courseNumber < node.data.courseNumber:  
 node = node.left  
 else:  
 node = node.right  
 print "course not found"  
  
function printCourse(course):  
 print course.courseNumber + ": " + course.name  
 if course.prerequisites is empty:  
 print "Prerequisites: None"  
 else:  
 print "Prerequisites: " + join(course.prerequisites, ", ")

In summary, using a binary search tree keeps all courses organized by their course number and allows advisors to quickly look up a course or list them in order without needing to sort again. This structure is efficient for both searching and displaying data once the file is loaded.