# CS 300 Pseudocode Document

## Function Signatures

// Vector pseudocode

Define a Course class with attributes: courseNumber, title, prerequisites  
Define a function to read data from file and populate the vector data structure:  
   function loadDataFromFile(filename)  
       open the file with filename for reading  
       if file not found or unable to open  
           print error message and exit  
       initialize empty vector courses  
       for each line in the file  
           parse the line to extract courseNumber, title, and prerequisites  
           if the line is not properly formatted (e.g., less than 2 parameters)  
               print error message and skip this line  
           else if prerequisites are provided  
               split prerequisites by comma  
               for each prerequisite in prerequisites  
                   if prerequisite not found in courses  
                       print error message and skip this line  
           create a new Course object with courseNumber, title, and prerequisites  
           add the Course object to the courses vector  
       close the file  
       return courses  
Define a function to search and print course information and prerequisites:  
   function printCourseInfo(courses, courseNumber)  
       found = false  
       for each course in courses  
           if course.courseNumber equals courseNumber  
               found = true  
               print "Course Number:", course.courseNumber  
               print "Title:", course.title  
               if course.prerequisites is not empty  
                   print "Prerequisites:"  
                   for each prerequisite in course.prerequisites  
                       print prerequisite  
               else  
                   print "No prerequisites"  
               break  
       if not found  
           print "Course not found"  
Main program:  
   courses = loadDataFromFile("filename.txt")  
   printCourseInfo(courses, "CSCI300")

// Hashtable pseudocode

Define a Course class with attributes: courseNumber, title, prerequisites  
Define a function to read data from file and populate the vector data structure:  
   function loadDataFromFile(CourseInfo)  
       open the file with filename for reading  
       if file not found and/or unable to open  
           print error message and exit  
       initialize empty vector courses  
       for each line in the file  
           parse the line to extract courseNumber, title, and prerequisites  
           if the line is not properly formatted (e.g., less than 2 parameters)  
               print error message and skip this line  
           else if prerequisites are provided  
               split prerequisites by comma  
               for each prerequisite in prerequisites  
                   if prerequisite not found in courses  
                       print error message and skip this line  
           create a new Course object with courseNumber, title, and prerequisites  
           add the Course object to the courses vector  
       close the file  
       return courses  
Define a function to search and print course information and prerequisites:  
   function printCourseInfo(courses, courseNumber)  
       found = false  
       for each course in courses  
           if course.courseNumber equals courseNumber  
               found = true  
               print "Course Number:", course.courseNumber  
               print "Title:", course.title  
               if course.prerequisites is not empty  
                   print "Prerequisites:"  
                   for each prerequisite in course.prerequisites  
                       print prerequisite  
               else  
                   print "No prerequisites"  
               break  
       if not found  
           print "Course not found"  
Main program:

   courses = loadDataFromFile("courseinfo.txt")  
   printCourseInfo(courses, "CSCI300")

// Tree pseudocode

Open the file containing course information  
If the file is successfully opened:  
    Initialize an empty vector to store course objects  
    For each line in the file:  
        Read the line  
        Tokenize the line using a comma as the delimiter  
        If the number of tokens is less than 2:  
            Print an error message indicating invalid file format (not enough parameters)  
            Skip to the next line  
        Else:  
            Extract course number, title, and prerequisites from the tokens  
            If there are prerequisites:  
                For each prerequisite in the list:  
                    Check if the prerequisite exists as a course in the vector  
                    If not:  
                        Print an error message indicating missing prerequisite  
                        Skip to the next line  
  
            Create a new course object with the extracted information  
            Add the course object to the vector  
    Close the file  
Else:  
    Print an error message indicating unable to open the file

Define a structure for a course node:  
    Course number  
    Course title  
    Prerequisites (vector of course node pointers)  
    Left child (pointer to left child node)  
    Right child (pointer to right child node)  
Define a function to print course information and prerequisites:  
    Input: Course node pointer (root of the tree)  
    Output: None (prints course information to the console)  
    If the input node is not null:  
        Print course number and title  
        If there are prerequisites:  
            Print "Prerequisites: "  
            For each prerequisite node in the vector of prerequisites:  
                Recursively call the print function with the prerequisite node as input  
        Else:  
            Print "No prerequisites"

//menu

Initialize empty data structure  
  
function load\_data\_structure(file):  
    Open file  
    Read lines from file  
    For each line:  
        Parse course information  
        Add course information to data structure  
    Close file  
  
function print\_course\_list():  
    if data structure is empty:  
        Print "Data structure is empty. Please load data first."  
    else:  
        Sort courses alphabetically by course number  
        For each course in sorted courses:  
            Print course information  
  
function print\_course(course\_number):  
    if data structure is empty:  
        Print "Data structure is empty. Please load data first."  
    else:  
        if course\_number exists in data structure:  
            Print course title  
            Print prerequisites for course  
        else:  
            Print "Course not found."  
  
function sort\_course\_info():  
    if data structure is empty:  
        Print "Data structure is empty. Please load data first."  
    else:  
        Sort courses alphabetically by course number  
        For each course in sorted courses:  
            Print course information  
  
menu\_loop:  
    Repeat until user chooses to exit:  
        Print menu options  
        Get user input for choice  
        if choice is "1":  
            Load Data Structure  
            Call load\_data\_structure function with file input  
        else if choice is "2":  
            Print Course List  
            Call print\_course\_list function  
        else if choice is "3":  
            Print Course  
            Get user input for course number  
            Call print\_course function with course number input  
        else if choice is "4":  
            Sort the course information  
            Call sort\_course\_info function  
        else if choice is "5":  
            Exit program  
        else:  
            Print "Invalid choice. Please select a valid option."

## Example Runtime Analysis

When you are ready to begin analyzing the runtime for the data structures that you have created pseudocode for, use the chart below to support your work. This example is for printing course information when using the vector data structure. As a reminder, this is the same pairing that was bolded in the pseudocode from the first part of this document.

| **Code** | **Line Cost** | **# Times Executes** | **Total Cost** |
| --- | --- | --- | --- |
| **for all courses** | 1 | n | n |
| **if the course is the same as courseNumber** | 1 | n | n |
| **print out the course information** | 1 | 1 | 1 |
| **for each prerequisite of the course** | 1 | n | n |
| **print the prerequisite course information** | 1 | n | n |
| **Total Cost** | | | 4n + 1 |
| **Runtime** | | | O(n) |