# CS 300 Pseudocode Document

## Function Signatures

## Pseudocode for Data Structure Operations:

## For Vector:

## Calculate the number of prerequisite courses for a given course.

## Print a sample schedule.

## Print course information based on a given course number.

## For Hashtable:

## Calculate the number of prerequisite courses.

## Print a sample schedule.

## Print course information based on a given course number.

## For Tree:

## Calculate the number of prerequisite courses.

## Print a sample schedule.

## Print course information based on a given course number.

## Design of File Handling and Course Object Creation:

## Define how the program opens the file, reads the data, parses each line, and checks for formatting errors.

## Create course objects that hold data from a single line in the input file.

## Menu Design Pseudocode:

## Create a menu with options:

## Load Data Structure: Load data into the chosen data structure.

## Print Course List: Print an alphanumerically ordered list of Computer Science courses.

## Print Course: Print the title and prerequisites for a selected course.

## Exit: Terminate the program.

## Alphanumeric Sorting and Display:

## Sort course information in alphanumeric order.

## Print the sorted list of Computer Science courses.

## Runtime Analysis:

## Analyze the runtime complexities (Big O) for various operations:

## Reading the file and creating course objects.

## Evaluate the worst-case runtime (Big O) for each data structure operation concerning file reading and course object creation.

## Advantages and Disadvantages Analysis:

## Evaluate the pros and cons of each data structure concerning the advisor's requirements.

## Consider factors like runtime complexity, memory usage, and suitability for the given tasks.

## Example Runtime Analysis

**m** is the number of lines in the file.

**n** is the number of courses.

**p** is the total number of prerequisites across all courses.

## Code: The pseudocode presented for printing course information using the vector data structure.

## Line Cost: Indicates the cost (time complexity) assigned to each line of code. In this analysis, the cost per line is considered to be 1.

## # Times Executes: Refers to the number of times each line or segment of code will execute.

## Total Cost: This column demonstrates the cumulative cost for each segment of code considering the number of times it executes.

## Runtime: Represents the overall runtime complexity of the code, expressed in Big O notation.

| **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) |