# 2 CS 300 Pseudocode Document

Part 1: Reading and Parsing the File

Part 3: Creating Course Objects and Storing in a Vector

Open the Course Information file

If the file is not open

Print "Error opening file"

Exit the program

Initialize an empty vector named courses

For each line in the file

Split the line into tokens separated by commas (courseNumber, title, and prerequisites)

If the number of tokens is less than 2

Print "Format error in the line" and the line content

Continue to the next line

Create a course object with courseNumber and title from the first two tokens

For each remaining token

Add the token to the course object's prerequisites list

Add the course object to the courses vector

Close the file

Part 2: Validating Prerequisites

For each course in courses

For each prerequisite in course's prerequisites list

If no course in courses has a courseNumber that matches the prerequisite

Print "Prerequisite error: " and the prerequisite " does not exist"

Part 4: Searching and Printing Course Information

Function printCourseInformation(courseNumber)

found = False

For each course in courses

If course's courseNumber matches the search courseNumber

Print "Course Number: " + course.courseNumber

Print "Title: " + course.title

If course's prerequisites list is not empty

Print "Prerequisites:"

For each prerequisite in course's prerequisites list

Print prerequisite

Else

Print "No prerequisites"

found = True

Break

If not found

Print "Course not found"

// Example usage

printCourseInformation("CS101")

This pseudocode outlines a basic structure for opening and reading a course information file, validating its format, creating course objects with prerequisites, storing them in a vector, and finally, searching this vector to print information about a specific course along with its prerequisites. Implementing this in a programming language would require defining the course object structure and handling the specifics of file I/O and string manipulation according to the language's syntax and libraries.

Top of Form

## Function Signatures

Below are the function signatures that you can fill in to address each of the three program requirements using each of the data structures. The pseudocode for printing course information, if a vector is the data structure, is also given to you below (depicted in bold).

// Vector pseudocode

int numPrerequisiteCourses(Vector<Course> courses, Course c) {

totalPrerequisites = prerequisites of course c

for each prerequisite p in totalPrerequisites

add prerequisites of p to totalPrerequisites

print number of totalPrerequisites

}

void printSampleSchedule(Vector<Course> courses) {

}

void printCourseInformation(Vector<Course> courses, String courseNumber) {

**for all courses**

**if the course is the same as courseNumber**

**print out the course information**

**for each prerequisite of the course**

**print the prerequisite course information**

}

// Hashtable pseudocode

int numPrerequisiteCourses(Hashtable<Course> courses) {

}

void printSampleSchedule(Hashtable<Course> courses) {

}

void printCourseInformation(Hashtable<Course> courses, String courseNumber) {

}

// Tree pseudocode

int numPrerequisiteCourses(Tree<Course> courses) {

}

void printSampleSchedule(Tree<Course> courses) {

}

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

}

## 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) |