# CS 300 Pseudocode Document

## 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) {

FOR all courses

print course name

IF course has prerequisites

FOR each prerequisite

print prerequisite

}

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) {

totalPrerequisites = Hashtable c

FOR each prerequisite p in totalPrerequisites

add prerequisites in Hashtable p to totalPrerequisites

print number of totalPrerequisites

}

void printSampleSchedule(Hashtable<Course> courses) {

FOR all key, value in pair

print key course name

IF value has prerequisites

FOR each prerequisite

print prerequisite

}

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

FOR all courses

IF course is the same as courseNumber

print course information

FOR each prerequisite of Hashtable(course)

print the prerequisite course information

}

// Tree pseudocode

int numPrerequisiteCourses(Tree<Course> courses) {

totalPrerequisites = left child and right child of Tree courses

FOR each prerequisite p in totalPrerequisites

add left and right child of node p in totalPrerequisites

Print totalPrerequisites

}

void printSampleSchedule(Tree<Course> courses) {

FOR all nodes as courses

print course name

IF course has left child

print left child as prerequisite

IF course has right child

print right child as prerequisite

}

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

FOR all nodes

IF course equals course number

print nodes information

ELSE course has left child

print left child as prerequisite for course information

ELSE course has right child

print right child as prerequisite for course info

}

## 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) |