**CS 300 Project One**

**Design pseudocode for reading the file and creating course objects:**

**Vector:**

function loadCoursesVector(){

open file "courses.txt"

while (not end of file){

line is equal to read line from file

if (line is formatted correctly){

course = parseCourse(line)

vector.append(course)

}

}

}

**Hash Table:**

function loadCoursesHashTable(){

open file "courses.txt"

while (not end of file){

line is equal to read line from file

if (line is formatted correctly){

course = parseCourse(line)

hashTable.insert(course.courseNumber, course)

}

}

}

**Tree:**

function loadCoursesTree() {

open file "courses.txt"

while (not end of file){

line is equal to read line from file

if (line is formatted correctly) {

course = parseCourse(line)

tree.insert(course)

}

}

}

**Design pseudocode for the menu:**

function menu(){

display "Choose an option:"

display "1. Load Data Structure"

display "2. Print Course List"

display "3. Print Course"

display "4. Exit"

choice equals user input

switch choice:

case 1:

call loadCourses<DataStructure>()

case 2:

call printCourseList<DataStructure>()

case 3:

courseNumber = user input

call printCourse<DataStructure>(courseNumber)

case 4:

}

**Design pseudocode to print the courses in alphanumeric order:**

**Vector:**

function printCourseListVector() {

sortedVector equals sort(vector)

for (course in sortedVector) {

display course

}

}

**Hash Table:**

function printCourseListHashTable(){

courseList equals hashTable.values()

sortedCourseList equals sort(courseList)

for (course in sortedCourseList){

display course

}

}

**Tree:**

function printCourseListTree(){

sortedCourses equals tree.inOrderTraversal()

for (course in sortedCourses){

display course

}

}

**Runtime Evaluation:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Loading courses | Printing List | Memory |
| Vector | O(n) | O(n log n) | O(n) |
| Hash Table | O(n) | O(n log n) | O(n) |
| Tree | O(n log n) | O(n) | O(n) |