**5-3 Project One Milestone Three: Tree Data Structure Pseudocode**

Struct Courses {

String cNum

String cTit

String cPre1

String cPre2

}

Struct Node {

Courses courses

Node \*left

Node \*right

}

**Object Creation**

void addNode(Node\* node, Courses course) {

IF root == nullptr {

Root = new Node(course)

}

Node\* curNode = root

IF curNode->right == nullptr {

curNode->right = new Node(course)

}

ELSE {

curNode = curNode->right

}

}

void insertCourses(Courses course) {

IF root == nullptr {

root = new Node(course)

}

ELSE {

addNode(root, course)

}

}

**Load Files**

void loadCourses(Node node, Courses course) {

For all files in the CSV file {

hash = 0

Use hashCode() FUNCTION to get course’s cNum hash value for a string

((s[0] \* 31 ^ (n – 1)) + (s[1] \* 31 ^ (n – 2)) + … + s(n – 1))

hash = course.cNum

IF root = NULL {

root = node(course)

node->left = NULL

node->right = NULL

}

ELSE {

WHILE root != NULL {

IF this.hash > 2000000000 {

curCor.course.cTit = parameter

}

ELSE IF parameter matches a prior node’s key {

IF curCor.course.cPre1 = NULL {

curCor.course.cPre1 = parameter

}

ELSE {

curCor.course.cPre2 = parameter

}

}

ELSE {

FOR new nodes {

curCor = root

IF node’s hash < curCor’s hash {

IF curCor->left = NULL {

curCor->left = node

curCor = NULL

}

ELSE {

curCor = curCor->left

}

ELSE {

IF curCor->right = NULL

curCor->right = node

curCor = NULL

}

ELSE {

curCor = curCor->right

}

}

}

Node->left = NULL

Node->right = NULL

}

}

}

}

Try {

FOR each course in courses {

Courses courses

Courses.cNum = file [i][1]

Courses.cTit = file [i][0]

Courses.cPre1 = file [i][2]

Courses.cPre2 = file [i][3]

Bst->insertCourses(courses)

}

}

Catch ERROR {

OUTPUT “Files not loaded correctly.”

}

}

**Print Course Information**

displayCourses(Courses courses) {

nNum = 1

FOR all courses {

IF course < 2 keys {

OUTPUT “Course has less than two parameters!”

}

ELSE {

OUTPUT nNum << “. “ << courses.cNum << “ “ << courses.cTit << “ “ courses.cPre1 << “ “ << courses.cPre2 << end newline

}

nNum++

}

}

(Example:

1. CSCI100 Introduction to Computer Science
2. CSCI101 Introduction to Programming in C++ CSCI100
3. CSCI200 Data Structures CSCI101
4. MATH201 Discrete Mathematics
5. CSCI300 Introduction to Algorithms CSCI200 MATH201
6. CSCI301 Advanced Programming in C++ CSCI101
7. CSCI350 Operating Systems CSCI300
8. CSCI400 Large Software Development CSCI301 CSCI350

)

void searchCourse(HashTable<Course> courses, String cNum) {

curCor = root

FOR all courses {

IF search = curCor->course.cNum {

return curCor

}

ELSE IF search < curCor->course.cNum {

curCor->left

}

ELSE IF search > curCor->course.cNum {

curCor->right

}

}

}