Project One Milestone Three

loadCourses(filename):

open file using filename

if file cannot be opened:

print error

return null

create empty binary search tree named courseTree

for each line in file:

For each line in file in file

Split line by “,” into tokens

If length of tokens < 2

Print error

Exit

For each prerequisite in tokens[2]

If prerequisite not in course

Print error

Exit

Course number = tokens[0]

Course title = tokens[1]

Prerequisites = tokens[2]

Create a course object

CourseNumber = course number

courseTitle = course title

prerequisites = prerequisites

add course object to courses with key = courseNumber

close file

return

return courseTree

class Course:

courseNumber

courseName

prerequisites (list)

createCourse(courseNumber, courseName, prerequisites):

newCourse = Course()

newCourse.courseNumber = courseNumber

newCourse.courseName = courseName

newCourse.prerequisites = prerequisites

return newCourse

function insertCourse(tree, course):

if tree is empty:

tree root = course

else:

insertHelper(tree root, course)

function insertHelper(node, course):

if courseNumber < node.courseNumber:

if node left is null:

node left = course

else:

insertHelper(node left, course)

else:

if node right null:

node right = course

else:

insertHelper(node right, course)

printCourses(tree):

inOrderTraversal(tree root)

inOrderTraversal(node):

if node not null:

inOrderTraversal(node left)

print course Number course name

if prerequisites

print prerequisites

inOrderTraversal(node right)