Jannatul Ferdush

CS 300

June 2, 2024

initialize hash table H

initialize vector V

function parse\_file(filename):

try:

open file with filename

for each line in file:

if line is not empty:

tokens = split line by delimiter ","

if tokens.length >= 2:

course\_number = tokens[0]

course\_title = tokens[1]

prerequisites = tokens[2:] if tokens.length > 2 else []

add\_course(course\_number, course\_title, prerequisites)

else:

print "Error: Insufficient parameters in line:", line

catch file open error:

print "Error: Unable to open file:", filename

function add\_course(course\_number, course\_title, prerequisites):

course = create\_course(course\_number, course\_title, prerequisites)

hash\_key = hash\_function(course\_number)

insert\_into\_hash\_table(H, hash\_key, course)

function create\_course(course\_number, course\_title, prerequisites):

course = new Course()

course.number = course\_number

course.title = course\_title

course.prerequisites = prerequisites

return course

function hash\_function(key):

return key mod table\_size

function insert\_into\_hash\_table(hash\_table, key, value):

index = hash\_function(key)

if hash\_table[index] is empty:

hash\_table[index] = value

else:

collision\_resolution(hash\_table, index, value)

function collision\_resolution(hash\_table, index, value):

// Implement collision resolution technique here

// For example, linear probing or chaining

function print\_course\_info():

for each slot in hash\_table:

if slot is not empty:

print "Course Number:", slot.number

print "Course Title:", slot.title

if slot.prerequisites is not empty:

print "Prerequisites:"

for each prerequisite in slot.prerequisites:

print prerequisite

else:

print "No prerequisites"