Augustus Mendy

CS-300-11373-M01

Southern New Hampshire University

May 18, 2024

4-3 Milestone Hash Table Data Structure Pseudocode

“START

//reading files

LOAD file

INITIALIZE call to open file

IF (return value is –1, the file is not found)

ELSE file is found

READ each line of file

IF (less than two values in a line)

RETURN error;

ELSE IF

Parameter has a third or more values continue first line with

courseNumber (prerequisite)

ELSE

Continue to read files

CLOSE file

PRINT all courses including prerequisites courseNumber at end of line

//Hash Table

CREATE hashTable class

INITIALIZE course <Node>nodes

CREATE courseId variable

CREATE courseTitle variable

CREATE prerequisite1 variable

CREATE prerequisite2 variable

CREATE prerequisiteCount variable

INITIALIZE prerequisiteCount equal to 0

CREATE temp item to hold values

CREATE current item that holds values with current pointer to next item

WHILE not end of file

LOOP through file

DECLARE unsigned int key

IF (node at key is not found)

INSERT new node at hash key %

ELSE IF (Node pointer to key equals UINT\_MAX)

Node pointer to key equals key

Node pointer to next equals nullptr

Node pointer to course = course

ELSE

WHILE (node -> next not equal to nullptr)

node equals node pointer to next.

new node (course, key)

//Print course information and prerequisites

VOID hashTable:: PrintALL()

FOR (unsigned int index equals 0; index less than tableSize; increment index by 1)

IF (node key is not equal to UINT\_MAX)

PRINT index node courseId, node prerequisite1, and node prerequisite2

WHILE (node next is not equal to nullptr)

node = node ->next

PRINT node key courseId, node prerequisite1, and node prerequisite2

RETURN”;

END