Daniel Schween

CS 300 SNHU

07/28/2024

**Project One Milestone Two**

**Pseudocode**

**LOAD** libraries and header.hpp files

**LOAD** file reader

**CREATE** a list structure, “courseList”, to store course data

**DEFINE “**tempList” and “tempCourse” to temporary hold course data

**CONSTRUCTOR Course{}:**

courseId = “”

courseName = “”

coursePrereq = “”

prereqCount = 0

**CLASS HashTable{}:**

-struct bucket: linkedList

Course

Key

Next pointer

+printAll()

+hash()

+courseList<> hashTable

**METHOD Parser():**

**OPEN** database file by invoking parser libraries

**READ** file data incrementing through the linked list.

**CHECK** the contents of each record and search for courseNumber, courseTitle, and if coursePrereq exists.

**CHECK/VALIDATE** file has been opened properly and course data can be read/written

**READ/WRITE** data by **LOOPING** row by row until the end of file:

**IF** first and second string are present

**CALL** hash passing the strings in tempList

**ADD** the first string to struct at courseId hash position

**ADD** the second string to struct at courseName hash position

**LOOP** through file the find a third string until coursePrereq is NULL

**INCREMENT** a variable, coursePrereq, for each prerequisite found

**RECORD** each prerequisite

**ELSE** NULL if no third string

**RETURN** tempList

**END**

**METHOD searchList():**

**CREATE** variable tempCourse reading to bucket location

**SET** tempCourse to the bucket hash location

**LOOP** through list for each course

**IF** tempCourse is the same as coursed

**SET** tempCourse equal to course

**RETURN** tempCourse

**END**

**METHOD printCourse():**

**SET** tempCourse equal to hash() of type bucket

**LOOP** through the chained buckets using tempCourse

**OUTPUT** courseId to a linked List found within tempCourse

**OUTPUT** courseName to linkedList found within tempCourse

**LOOP** through course data from 0 to prereqCount

**FOR** each course in coursePrereq

**CALL** printCourse() passing coursePrereq

**END**

**METHOD validateList():**

**CREATE** tempCourse of type bucket

**CREATE** bool variable “valid” and set to true

**FOR** each course

**IF** “valid” is false

**BREAK**

**WHILE** tempCourse is not NULL

**LOOP** from 0 to prereqCount

**SET** tempCourse equal to coursePrereq

**IF** tempCourse, courseId, is empty

**SET** “valid” to false

**RETURN** “valid”

**END**

**MAIN():**

**CREATE** new list named “courseList”

**GET** CSV file path from user

**IF** no data is passed, use default location

**CALL Parser()** passing CSV file path

**CALL validateList()**  passing courseList

**CALL searchList()**  to **LOOP** through course data

**VALIDATE** courseId, courseTitle, and coursePrereq

**GET** user value to search and store in variable “userSearch”

**CALL printCourse()** passing “userSearch”

**STORE** 1st and 2nd string in courseList

**STORE** 3rd string in validateList()

**END**