Corey Gaspar

3/27/2025

**Project One Milestone Two – Hash Table Pseudocode**

**START**

**LOAD** libraries and CSV parser headers

**DEFINE** a struct that will hold course data

**struct Course {}**

**STRING** courseID

**STRING** courseName

**INT** preCount

**STRING** preList

**FUNCTION Course()** {

**SET** courseID to empty

**SET** courseName to empty

**SET** preCount to 0

**SET** preList to empty

}

**CLASS** HashTable():

**STRUCT** Bucket:

**ATTRIBUTE** course

**ATTRIBUTE** key

**ATTRIBUTE** nextPointer

**END** struct

**FUNCTION** hash

**FUNCTION** printAllCourses

**CREATE** hashTable list

**Main()**

**PRINT** “Enter CSV file path (or press enter to use default path):”

**READ** filePath

**IF** no file path is passed, use default location

**CALL** txtParser() process the CSV file and load data into the new list courseList

**CALL** validateList() to validate the data in the list courseList

**GET** user search input and **STORE** in userSearch

**CALL** printCourse() to display course information

**END**

**FUNCTION txtParser (STRING)**

**OPEN** file path using the parser libraries loaded at the beginning

**CREATE** list tempList

**WHILE** not at the end of the file:

**READ** currentRow **FROM** CSV file

**SPLIT** currentRow **INTO** columns

**IF** first and second columns **ARE NOT** empty:

**CALL** hash(columns[0])

**CREATE** struct newCourse

**SET** courseId to the first column

**SET** courseName to the second column

**SET** preCount to 0

**SET** preNames to empty

**LOOP** through rest of the columns until there are no more columns left

**ADD** preCount to struct at preCount

**ADD** preList to struct at prelist

**ADD** newCourse to tempList

**RETURN** tempList

**END**

**FUNCTION searchList(STRING)**

**CREATE** tempCourse of type bucket

**SET** tempCourse equal to the bucket found at hash location of courseId

**WHILE** tempCourse **IS NOT** empty:

**IF** courseId in tempCourse matches courseId, return tempCourse

**ELSE IF** courseId not found:

**RETURN** null;

**END**

**FUNCTION printCourse(STRING)**

**SET** tempCourse equal to bucket at hash location of courseId

**WHILE** tempCourse is not null:

**PRINT** courseId and courseName from tempCourse

**FOR EACH** prereq in preList:

**CALL** printCourse() for the prerequisite

**MOVE** to next bucket

**END**

**FUNCTION validateList()**

**SET** initial value of isValid to True

**FOR EACH** course in courseList:

**IF** isValid is False, stop checking for course

**FOR EACH** preList in course

**SET** tempCourse to searchList

**IF** tempCourse is empty, set isValid to False and stop checking

**RETURN** isValid

**END**