Southern New Hampshire University
Joshua Wozny
joshua.wozny@snhu.edu
January 30, 2023
CS-300 Week 4: Milestone

PSEUDOCODE

Parser(string csvPath) csvPath is the name of the file to import with its relative path included

- Open a file stream
- Initialize a vector to hold each line from the input file
- While there are still lines in the file
    - add each line to the vector of input lines
- Parse the first element in the vector as the headers
    - separate the line into separate elements with each element separated by a comma into a vector to store header elements
        - add each element into the header vector
        - add the header element to the vector of rows as the first element
- Iterating over the vector of input lines, starting at the second element
    - for each line, separate the line into separate elements with each element separated by a comma into a vector of row elements
    - add the vector of row elements to the vector of rows
- Iterating over the vector of rows
    - count the number of elements in the row vector.
    - If number of elements < 2, return an error,
    - otherwise, iterating over the vector of rows again
        - add value in first element into a vector of course numbers
    - Iterating over the vector of rows again
        - for each element starting at element three
        - if value is in the vector of course numbers is in vector of course numbers
        - List is valid, return the vector of rows
    - List is invalid, return an error

Process(Parser(string csvPath)) process the parsed list

- Initialize vector<LinkedList>(tableSize) courseLibrary, where tableSize is the number of “buckets” used for hashing.

- AddCourse()

AddCourse(vector row) a row is an element of the vector of rows returned from Parser (except element one which contains the header information.

- Initialize a Course object with the following elements
    - string courseNumber = first element of row
    - string courseName = second element of row
    - Initialize a vector<string> prerequisites =
        - for each element of row after 2, if it exists, add to prerequisites
- key = hash(course) using an appropriate hash function that divides courses into "buckets" as equally as possible between the number of buckets defined in tableSize
- Add the Course object to LinkedList = courseLibrary.at(key)

Course SearchCourse(string courseNumber)

- Iterate over each LinkedList in courseLibrary
    - if LinkedList is empty, continue on to the next element of courseLibrary
    - otherwise, loop over each node of the LinkedList, if the course number of the current Course = courseNumber
        - return the current Course
    - otherwise continue to the next element of courseLibrary and loop over each node of the LinkedList, if the course number of the current Course = courseNumber
        - return the current Course
    - continue until you reach the end of courseLibrary, if courseNumber not found then return an empty Course object

DisplayCourse(SearchCourse(string courseNumber))

- Print a header: Course Number Course Name Pre-requisites
- Print on one line course.courseNumber course.courseName
- Iterate over each element of course.prerequisites and print each element on the same line separated by commas