Millstone: Pseudocode for ABCU Project One

Jason Kremhelmer

Southern New Hampshire University

CS300: Analysis and Design

David Ostrowski

3/31/2024

# Pseudocode

**Load** text parsing libraries and headers

**Define** a struct to hold course data

**struct Course {}**

*courseID*

*courseName*

*preCount*

*prelist*

Course() (constructor) {courseID = courseName = ”; preCount = 0; preList = ”}

**Class HashTable{}**

-struct *bucket*

*Course*

key

*next* pointer

+*hash()*

*+printAll()*

+List<> *hashTable*

**Main()**

**Create** new List named *courseList* of the struct-type **CourseMap**

**Get** CSV file path from user

**If** no data passed use default location

**Call** **txtParser**() passing CSV file path

**Call** **validateList**() passing *courseList*

**Get** user value to search for and **Store** in *userSearch*

**Call** **printCourse()** passing *userSearch*

**End**

**txtParser (String)**

**Open** file found at the path in *String* by invoking parser libraries

**Loop** row by row until end of file (eof)

**If** first and second string are present

**Call** hash passing the first string

**Add** to struct at hash position within *tempList*

**Add** the first String to struct at *courseID*

**Add** the second String to Struct at *courseName*

**Loop** until file handler has no value in a column (indicates no more prerequisite)

**Increment** a variable named *preCount* for each prerequisite found

**Concatenate** a localString named *preNames* for each prerequisite

**Add** *preCount* to struct at *preCount*

**Add** *preNames* to struct at *preList*

**Return** *tempList*

**End**

**searchList(String)**

**Create** *tempCourse* of type **bucket**

**Set** *tempCourse* to the bucket at the hash location of *String*

**Loop** through list For Each Course

**If** *String* is the same as *courseID*

**Set** *tempCourse* to Course

**Return** *tempCourse*

**End**

**printCourse(String)**

**Create** *tempCourse* of type **bucket**

**Set** *tempCourse* equal to **hash (**String**)**

**Loop** through all chained buckets at *tempCourse*

**Output** *courseID* in Course struct found within *tempCourse* to console

**Output** *courseName* in Course struct found within *tempCourse* to console

**Loop** 0 to *preCount*

**For each** *Course* in *preList*

**Call** p**rintCourse**() passing *preList*

**End**

**validateList**()

**Create** *tempCourse* of type **bucket**

**Create** variable *valid* and **Set** to True

**For Each** **Course**

**If** *valid* is False break

**While** *tempCourse* next is not null

**Loop** 0 to *preCount*

**Set** *tempCourse* equal to **searchList(***preList* token**)**

**If** *tempCourse* courseID is empty Set *valid* to False

**Return** valid

**End**

**int Hash(key)**

# to be decided; how do we want to hash the String *courseID*

# Hash function needs to be cognizant that the first 4 letters, e.g., CSCI, will appear

# often for a given major

# Hash also needs to be aware that the last three numerals may not be unique, e.g.,

# CSCI100 is distinct from MATH100

**return** hash of key