# Thomas Martin

# CS-300-H2990 DSA: Analysis and Design 23EW2

# 4-3 Milestone: Hash Table Data Structure Pseudocode

# Southern New Hampshire University

**November 23, 2023**

DECLARE void loadCourses(string csvPath, HashTable \*hashTable)

//passes in the file path and instance of hashTable

//will be used in the filestream to run various methods on the passed in file.

DECLARE filestream variable courseList

If stream courseList;

OPEN file

courseList.open(csvPath);

//check for errors opening the file

WHILE courseList is good

//used to store data as filestream parses the file

DECLARE string line;

//passes in filestream name, variable to store data and delimiter to let the parser know to stop at that line and store data.

INVOKE method getline(courseList, line, ',');

PRINT line

Course Object Pseudocode

//data provided by csv file previously loaded

DEFINE struct to hold course data

Struct Course

//stores the course id and will be used as the key in hash table

DECLARE variable courseID;

// stores the name of the course

DECLARE variable title;

//stores the first prerequisite

DECLARE variable preReq1;

//stores the second prerequisite

DECLARE variable preReq2;

//variable to store the number of prerequisites

DECLARE variable preReqCount;

//method that initializes the preReqCount

DECLARE method Course()

//method that initializes the preReqCount

INITIALIZE preReqCount equal to 0;

//defines structure of a hash table

DEFINE class HashTable

DEFINE private data members of class HashTable

PRIVATE:

//a structure to hold courses

DEFINE Struct Node to hold courses

DECLARE Course courses;

//will be used store hashed courseID

DECLARE unsigned int key;

//pointer to the next node

DECLARE Node \*next;

// Default constructor

DEFINE Node() method

//stores unsigned int max value to key

INITIALIZE “key” to UINT\_MAX;

//sets the next nodes pointer to null

INITIALIZE “next” to null pointer;

// Define Node method that passes in a course as a parameter and uses single colon to inherit from Node()

DEFINE Node (Course aCourse) : Node ()

INITIALIZE “course” to aCourse;

// Define Node method with two parameters (course, aKey) passed in as parameters and uses single colon to inherit from Node (Course course)

DEFINE Node (Course course, unsigned int aKey) : Node(course)

INITIALIZE “key” to aKey;

DECLARE a vector of courses (uninitialized)

DECLARE integer “tableSize” and SET to DEFAULT\_SIZE;

DECLARE integer method “hash” and with key as parameter;

DEFINE public data members

PUBLIC:

DECLARE method HashTable();

DECLARE method PRINTALL ()

DEFINE method hash() that takes in key as a parameter

DECLARE integer “hashedKey”;

INITIALIZE “hashedKey” to EVALUATE expression key MODULO (%) tableSize;

RETURN hashedKey

Print Course Information Pseudocode

DEFINE method of type void HashTable::PrintAll()

ITERATE through hash table

DEFINE FOR loop

DECLARE and INITIALIZE loop iterator “i” and SET to 0;

SET “i” LESS THAN size of the hash table;

INCREMENT “i” by one;

DECLARE pointer Node \*currNode

SET currNode equal to reference pointer courses.at(ith) position;

IF currNode key IS NOT equal to UINT\_MAX

PRINT index “i” << currNode courseID << currNode preReq1 << currNode preReq2

WHILE currNode next IS NOT null pointer

SET currNode to next node

PRINT currNode key << currNode courseID << currNode preReq1 << currNode preReq2

RETURN;