**Project One Milestone Two**

Hash Table Structure – Data Parser Pseudocode

void parseContent(String csvFile) {

**Open the CSV file passed into the function.**

**Loop through each row in the CSV file {**

**If row has at least two parameters (courseNumber, name) {**

**If row has no prerequisites {**

**Push current row’s course to the end of the vector.**

**}**

**Else {**

**If prerequisite(s) is already stored in the vector {**

**Push current row’s course to the end of the vector.**

**}**

**Else {**

**Throw error that prerequisite is not found in vector.**

**}**

**}**

**Else {**

**Throw error that row is missing necessary parameters.**

**}**

**}**

Hash Table Structure – Data Loading Pseudocode

struct Node {

Course course;

unsigned int key;

Node \*next;

}

struct Course {

string courseNumber (unique identifier)

string name

vector<courseNumber> prerequisites

}

vector<Node> loadCourses(String csvPath) {

**Define a vector data structure to hold a collection of nodes**

**Initialize the CSV parser using the given csvPath parameter**

**If file is open**

**Loop to read rows of the CSV file**

**Use the modulo hash to determine the key value to insert each bid into**

**(courseNumber % Table Size)**

**If Key value’s node is empty**

**Create a Linked List to store each Course**

**Insert Course as Linked List head and tail**

**Else**

**Append Course to end of Linked List**

**Close File**

**If File Size is 0**

**Throw a “No data in File” error**

**Else**

**Throw a “Failed to open file” error**

**Return the vector of nodes.**

}

Hash Table Structure – Search and Print Pseudocode

void searchCourse(Vector<Course> nodes, String courseNumber) {

**Use the courseNumber to generate a hash key**

**For all nodes**

**If hash key matches node key**

**Iterate through node’s list of courses**

**If courseNumber matches courseNumber of current list item**

**Print out the course information**

**For each prerequisite of the course**

**Print the prerequisite course information**

}