**Project One Milestone Three**

Binary Search Tree 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.**

**}**

**}**

Binary Search Tree Structure – Data Loading Pseudocode

struct Node {

Course course;

Node \*left;

Node \*right;

}

struct Course {

string courseNumber (unique identifier)

string name

vector<courseNumber> prerequisites

void loadCourses(String csvPath, BinarySearchTree\* bst) {

**Define a BST 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**

**If Root node is empty**

**Insert Course as Root node**

**Else**

**If courseNumber is smaller than current node’s (starting at root) courseNumber**

**If current node has no left child**

**Insert Course as new left child**

**Else**

**Recurse down the left node**

**Else If courseNumber is larger than current node’s courseNumber**

**If current node has no right child**

**Insert Course as new right child**

**Else**

**Recurse down the right node**

**Close File**

**If File Size is 0**

**Throw a “No data in File” error**

**Else**

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

}

Binary Search Tree Structure – Search and Print Pseudocode

Course searchCourse(String courseNumber) {

**Set current node equal to root of the BST**

**While current isn’t an empty node**

**If current’s courseNumber is equal to the input courseNumber**

**Return Course**

**Else If current’s courseNumber is larger than the input courseNumber**

**Set current node to current’s left child (to traverse left in tree)**

**Else (If current’s courseNumber is smaller than the input courseNumber)**

**Set current node to current’s right child (to traverse right in tree)**

**Return Empty Course**

}