Kristle Morales Figueroa

CS-300 Project One Milestone Three

1. **Pseudo code for opening, reading, parsing through a file:**

Open file

If the file is not open

Print “File Not Open”

End the program

While not at the end of the file

Read each line and copy into a string

Separate string into a list of items by spaces

If less than 2 spaces are present in the string

Print “Error”

Skip to next line

Else add string to “courses”

Close file

1. **Pseudo code for creating course objects and store them in a data structure**

**Create** a structure **Courses** to contain the courses information

**Define** identifier and data type string courseNumber, name, prerequisite1, prerequisite2

**Create** a Binary Search Tree structure

**Create** a new node

**Define** the **Courses** element

**Define** a left and right node

**Initialize** the left and right node to nullptr using the constructor

**Declare** private functions

**Create** a new node called *root*

**Create** a function for adding an object

**Create** a function for removing an object

**Create** a function for displaying all objects

**Declare** public functions

**Call** methods for searching

**Call** methods for inserting

**Call** methods for removal

**Define** *root* as nullptr in the default constructor

BinarySearchTree::Insert(Courses course)

If the root is nullptr

Assign root to a new course node

Else add a node with the root and course

BinarySearchTree::Remove(Courses course)

If the root is nullptr

Return node

If the root is not empty go down the left side of the tree to find a match

If a match is found remove it

If the root is not empty go down the right side of the tree to find a match

If a match is found remove it

Else the tree is a leaf node

If the left node is empty and the right node is empty

Delete the node

If the left node is not empty and the right node is empty

Create a temp node

Assign temp to the left node

Delete the node

If the left node is empty and the right node is not empty

Create a temp node

Assign temp to the right node

Delete the node

Else more than one child is present

Create a temp node assigned to the right node

Loop through the left side of the tree to the not empty node

Assign the node to temp

Remove the node

Return

1. **Pseudo code for search and print: (*This pseudo code was provided on the Supporting Materials Document.*)**

BinarySearchTree::Search

Create a new node current and assign it to root.

Loop through until a match to current is found

If a match is found

Return current

Else go down the left side of the tree

If a match is found

Assign current to the found node

Else go down the right side of the tree

If a match is found

Assign current to the found node

If no node is found after going through the list

Assign the course element

Return course