Project One: ABCU

Jason Kremhelmer

Southern New Hampshire University

CS300: Analysis and Design

David Ostrowski

4/14/2024

Pseudocode for ABCU Project One

**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 BinaryTree{}**

-struct *Node*

*Course*

*right* pointer

*left* pointer

-*root*

*+printCourse()*

+*BinaryTree()*

**Main()**

**Create** new BinaryTree named *courseTree* of the struct-type **Course**

**Get** CSV file path from user

**If** no data passed use default location

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

**Call** **validateList**() passing *courseTree*

**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

**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 **Node**

**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 **root**

**Loop** until *tempCourse* is Null

**If** the Node at tempCourse contains a *bidId* equal than to *String*

**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*

**If** the Node at *tempCourse* contains a *courseID* less than to *String*

**Set** *tempCourse* equal to the left Node

**If** the Node at *tempCourse* contains a *courseID* greater than to *String*

**Set** *tempCourse* equal to the right Node

**End**

**validateList**()

**Create** *tempCourse* of type **Node**

**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**