Corey Gaspar

4/2/2025

**Project One Milestone Three Pseudocode**

**START**

**LOAD** libraries and CSV parser headers

**DEFINE** a struct that will hold course data

**struct Course {}**

**STRING** courseID

**STRING** courseName

**INT** preCount

**STRING** preList

**FUNCTION Course()** {

**SET** courseID to empty

**SET** courseName to empty

**SET** preCount to 0

**SET** preList to empty

}

**Main()**

**CREATE** BinaryTree courseTree

**PRINT** "Enter the CSV file path (or press Enter to use default):"

**READ** filePath **FROM** user input

**IF** filePath is empty **THEN**

**SET** filePath = "default\_location.csv" // Use default file path

**END IF**

**CALL** txtParser(filePath)

**CALL** validateList(courseTree)

**PRINT** "Enter the course ID to search for:"

**READ** userSearch **FROM** user input

**CALL** printCourse(userSearch)

**END**

**FUNCTION txtParser(STRING)**

**OPEN** file at filePath

**CREATE** tempList (to store all Course objects)

**WHILE** not end of file

**READ** line **FROM** file

**SPLIT** line **INTO** courseID, courseName, prerequisites (if available)

**IF** courseID **AND** courseName are present **THEN**

**CREATE** newCourse

**SET** newCourse.courseID = courseID

**SET** newCourse.courseName = courseName

**SET** preCount = 0

**SET** preNames = ""

**WHILE** there are more prerequisites in the line **DO**

**INCREMENT** preCount

**CONCATENATE** prerequisite to preNames

**SET** newCourse.preCount = preCount

**SET** newCourse.preList = preNames

**ADD** newCourse to tempList

**RETURN** tempList

**END**

**FUNCTION searchList(STRING)**

**CREATE** new node called tempCourse

**SET** tempCourse to the node at courseID

**LOOP** through list of courses:

**FOR EACH** course in the list:

**IF** courseID matches courseID of course:

**SET** tempCourse = current course

**RETURN** tempCourse

**END**

**FUNCTION printCourse(STRING)**

**CREATE** Node tempCourse

**SET** tempCourse = starting point of tree (root)

**LOOP** until tempCourse = null

**IF** courseId in tempCourse = courseId

**PRINT** courseID from the Course struct in tempCourse

**PRINT** courseName from the Course struct in tempCourse

**LOOP** from 0 to preCount

**FOR EACH** course in preList:

**CALL** printCourse()

**ELSE** **IF** courseId in tempCourse < courseId

**SET** tempCourse = left child node

**ELSE IF** courseId in tempCourse > courseId

**SET** tempCourse = right child node

**END**

**FUNCTION validateList()**

**CREATE** new Node tempCourse

**CREATE** variable isValid and set it to True

**FOR EACH** course in the list

**IF** isValid = false

**BREAK** loop

**WHILE** next tempCourse is **NOT** null

**LOOP** through preList

**FOR EACH** preList

**SET** tempCourse = searchList

**IF** tempCourse courseId is empty

**SET** isValid to False

**BREAK** loop

**RETURN** isValid

**END**