**Sabastian Fasnao**

**CS-300**

**January 25, 2025**

**Module 3 Milestone: 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 = “”}

Main()

* Create a new list named courseList of type Course.
* Prompt the user to input the path to a CSV file.
  + If no path is provided, use the default file location.
* Call txtParser() and pass the CSV file path.
* Call validateList() and pass courseList.
* Prompt the user to input a search value and store it in userSearch.
* Call printCourse() and pass userSearch.

End

txtParser(String)

* Create a local list named tempList.
* Open the file at the specified path using parsing libraries.
* Iterate through the file row by row until the end of the file (EOF).
  + If both the first and second strings are present:
    - Assign the first string to courseID in the struct.
    - Assign the second string to courseName in the struct.
    - Loop through the remaining columns until no more prerequisites are found:
      * Increment a variable preCount for each prerequisite encountered.
      * Append each prerequisite to a local string preNames.
    - Assign preCount to the struct's preCount.
    - Assign preNames to the struct's preList.
* Return tempList.

End

searchList(String)

* Create a temporary variable tempCourse of type Course.
* Iterate through the list of courses:
  + If the String matches courseID, set tempCourse to the current course.
* Return tempCourse.

End

printCourse(String)

* Create a temporary variable tempCourse of type Course.
* Set tempCourse to the result of searchList(String).
* Output courseID to the console.
* Output courseName to the console.
* Loop from 0 to preCount:
  + For each course in preList, call printCourse() with the current prerequisite.

End

validateList()

* Create a temporary variable tempCourse of type Course.
* Create a variable valid and set it to True.
* Iterate through each course:
  + If valid is False, exit the loop.
  + Loop through prerequisites from 0 to preCount:
    - Set tempCourse to the result of searchList(preList token).
    - If the courseID of tempCourse is empty, set valid to False.
* Return valid.

End