CS 300 Project Two Guidelines and Rubric

Competency

In this project, you will demonstrate your mastery of the following competency:

- Develop code using algorithms and data structures to solve basic programming problems

Scenario

The academic advisors in the Computer Science department at ABCU are very happy with the planning and design you have completed for the advising assistance software. Now, they would like you to write the code for the application so the department advisors can use it to help when they talk with students.

Directions

Complete all of your coding in the integrated development environment (IDE). Additional references on the use of this IDE are linked in the Supporting Materials section. As you begin coding, use the data structure that you recommended in Project One to complete the following actions:

1. **Input: Design code to correctly read the course data file**. The program you submit should be a command-line program written in C++. The program should prompt the user to ask for the file name that contains the course data and read that file into course objects that are stored in your chosen data structure. Your data structure will hold course objects.
2. **Menu: Design code to create a menu that prompts a user for menu options**. The menu should include the following options:
    a. Option 1: Load the file data into the data structure. Note that before you can print the course information or the sorted list of courses, you must load the data into the data structure.
    b. Option 2: Print an alphanumeric list of all the courses in the Computer Science department.
    c. Option 3: Print the course title and the prerequisites for any individual course.
    d. Option 9: Exit the program.
3. **Loading Data Structure: Develop working code to load data from the file into the data structure**.
4. **Course List: Develop working code to sort and print out a list of the courses in the Computer Science program in alphanumeric order,** including all math courses. To print out a course list, use the pseudocode you created previously to guide your work. Then create code that will allow advisers to print a course list in alphanumeric order. Remember that this code should do the following actions:
    a. Sort the course information alphanumerically from lowest to highest.
    b. Print the sorted list to a display.

5. **Course Information: Develop working code to print course information.** This code should allow users to look up a course and print out information about its title and prerequisites. Your program must prompt the user to enter the course number. You will then print the name of the course and the prerequisite course numbers and titles. See Project Two Sample Program Output in the Supporting Documents section.
6. **Industry Standard Best Practices: Apply industry standard best practices in code design**. Your program should display an error message when user input does not fall within the parameters. You should also use in-line comments and appropriate naming conventions to enhance readability and maintainability.

What to Submit

To complete this project, you must submit the following item:

Advising Assistance Program

Submit your ProjectTwo.cpp C++ code file. Ideally, all of your code should be in a single CPP file with all of the necessary functions working correctly without any CSVParser or other header files being used. Do not zip or compress your submission. Make sure the code compiles and runs and has been thoroughly tested.

Supporting Materials

The following resources may help support your work on the project:

Reading: Course Information

This document outlines the courses and pathway you will be designing for.

File: ABCU Advising Program Input

This file contains all of the course information you will need to run your program. The text is written in comma-separated values for the fields that include course number, course title, and prerequisites. The course number and title will be on every line in the file. A course may have no prerequisites, or it may have one or more. A prerequisite will be listed with its course number.

Reading: Project Two Sample Program Output

This file shows an example of the kind of output you would expect from the program you are designing.

Project Two Rubric

Criteria	Exemplary (100%)	Proficient (85%)	Needs Improvement (55%)	Not Evident (0%)	Value
Input	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner	Designs code to correctly read a data file	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include taking care of missing files	Does not attempt criterion	15

Criteria	Exemplary (100%)	Proficient (85%)	Needs Improvement (55%)	Not Evident (0%)	Value
Menu	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner	Creates a menu that prompts a user for appropriate input	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include ensuring the output matches exactly what is provided in the sample	Does not attempt criterion	15
Loading Data Structure	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner	Develops code to load file data into the data structure	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include displaying the prerequisites correctly or functioning correctly with multiple input files	Does not attempt criterion	20
Course List	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner	Develops the appropriate algorithm to print out a sorted list of data from a data structure	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include properly sorting the list of courses	Does not attempt criterion	20
Course Information	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner	Develops the appropriate algorithm that looks up a course using an appropriate data structure	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include properly displaying the course information	Does not attempt criterion	20

Criteria	Exemplary (100%)	Proficient (85%)	Needs Improvement (55%)	Not Evident (0%)	Value
Industry Standard Best Practices	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner	Applies industry standard best practices in code design	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include adding more comments, naming all variables by convention, or displaying an error message when user input does not fall within parameters	Does not attempt criterion	10
				Total:	100%