**Competencies**

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

* Apply non-coding development methodologies for outlining an algorithmic design
* Evaluate complex data structures that solve a given problem using advanced algorithmic designs

**Scenario**

The academic advisors in the Computer Science department at ABCU are very happy with the pseudocode you completed. You are now prepared to move forward with expanding the pseudocode to directly respond to the two items advising hopes to accomplish with this program. Remember, your program will need to do the following:

1. Print a list of all the Computer Science courses in alphanumeric order.
2. For a given course, print out its title and prerequisites.

You will be writing pseudocode to address each of advising’s requirements, and you will do so for each of the data structures you already started to explore in the previous assignments (vector, hash table, and tree). Then you will perform a runtime analysis to determine which data structure will be the best to use when you begin coding in the next project.

**Directions**

In previous milestones, you wrote a description for the Course object that will be stored in different data structures. To be able to determine the running time of each of those data structures in this application, you will need to finish writing all the pseudocode for the rest of the code and perform a Big O analysis.

**Pseudocode**

1. **Resubmit pseudocode from previous pseudocode assignments and update as necessary**. In the previous assignments, you created pseudocode for each of the three data structures (vector, hash table, and tree). Be sure to resubmit the following pseudocode for each data structure.
   1. Design pseudocode to define how the program opens the file, reads the data from the file, parses each line, and checks for formatting errors.
   2. Your pseudocode should show how to create course objects, so that one course object holds data from a single line from the input file.
   3. Design pseudocode that will print out course information and prerequisites.
2. **Create pseudocode for a menu**. The menu will need to perform the following:
   1. Load Data Structure: 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.
   2. Print Course List: This will print an alphanumerically ordered list of all the courses in the Computer Science department.
   3. Print Course: This will print the course title and the prerequisites for any individual course.
   4. Exit: This will exit you out of the program.
3. **Design pseudocode that will print out the list of the courses in the Computer Science program in alphanumeric order.** Continue working with the Pseudocode Document linked in the Supporting Materials section. Note that you will be designing for the same three data structures that you have been using in your previous pseudocode milestones (vector, hash table, and tree). This time you will create the final pieces of pseudocode that you will need for ABCU’s advising program. To complete this part of the process, do the following:
   1. Sort the course information by alphanumeric course number from lowest to highest.
   2. Print the sorted list to a display.

**Evaluation**

1. **Evaluate the run-time and memory of data structures that could be used to address the requirements**. In a previous assignment, you created pseudocode to do the following:
   1. Define how the program opens the file, reads the data from the file, parses each line, and checks for formatting errors.
   2. Show how to create course objects, so that one course object holds data from a single line from the input file.

Using this pseudocode written for the previous assignments, analyze the worst-case running time of each, reading the file and creating course objects, which will be the Big O value. This should not include the pseudocode written for the menu or the sample schedule above. To do this, do the following:

* 1. Specify the cost per line of code and the number of times the line will execute. Assume there are n courses stored in the data structure.
  2. Assume the cost for a line to execute is 1 unless it is calling a function, in which case the cost will be the running time of that function.

1. Based on the advisor’s requirements, analyze each data structure (vector, hash table, and tree). **Explain the advantages and disadvantages of each structure in your evaluation.**
2. Now that you have analyzed all three data structures, **make a recommendation for which data structure you will plan to use in your code**. Provide justification for your recommendation, based on the Big O analysis results and your analysis of the three data structures.

**What to Submit**

To complete this project, you must submit the following:

**Pseudocode and Runtime Analysis**  
Your submission should be formatted in a double-spaced, 1–2 page Word document that includes your completed pseudocode, your runtime analysis in a chart, and your analysis of both the advantages and disadvantages for each structure.

**Supporting Materials**

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

[Course Information](https://learn.snhu.edu/content/enforced/1160013-CS-300-H7580-OL-TRAD-UG.22EW1/course_documents/CS%20300%20Course%20Information.pdf?_&d2lSessionVal=NyYpLMW7HTRHzQxu0M2A1JaSC&ou=1160013)  
This document outlines the courses and pathway you will be designing for.

[Pseudocode Document](https://learn.snhu.edu/content/enforced/1160013-CS-300-H7580-OL-TRAD-UG.22EW1/course_documents/CS%20300%20Pseudocode%20Document.docx?_&d2lSessionVal=NyYpLMW7HTRHzQxu0M2A1JaSC&ou=1160013)  
This document provides sample pseudocode and a runtime analysis that you will use to support your work in this project. You have already completed work in this document during previous milestones. During this project, you should make updates and additions to your pseudocode based on the directions and rubric. Note that the original Pseudocode Document is only provided again for reference.

| **Project One Rubric** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Criteria** | **Exemplary (100%)** | **Proficient (85%)** | **Needs Improvement (55%)** | **Not Evident (0%)** | **Value** |
| **Pseudocode for Prerequisites** | Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner | Designs pseudocode that opens and reads from a file, creates an object to store the data, and prints data based on the requirements | Shows progress toward proficiency, but with errors or omissions | Does not attempt criterion | 10 |
| **Pseudocode for Menu** | Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner | Designs pseudocode that provides users with required menu functionality | Shows progress toward proficiency, but with errors or omissions | Does not attempt criterion | 10 |
| **Pseudocode for Course list** | Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner | Designs pseudocode with appropriate algorithms to print out an sorted list from a data structure | Shows progress toward proficiency, but with errors or omissions | Does not attempt criterion | 20 |
| **Run-Time Evaluation** | Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner | Evaluates the run-time and memory of data structures that could be used to address requirements | Shows progress toward proficiency, but with errors or omissions | Does not attempt criterion | 20 |
| **Advantages and Disadvantages** | Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner | Explains the advantages and disadvantages of various data structures based on requirements | Shows progress toward proficiency, but with errors or omissions | Does not attempt criterion | 15 |
| **Recommendation** | Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner | Provides a justification for a recommended data structure, based on a given set of requirements | Shows progress toward proficiency, but with errors or omissions | Does not attempt criterion | 20 |
| **Articulation of Response** | Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner | Clearly conveys meaning with correct grammar, sentence structure, and spelling, demonstrating an understanding of audience and purpose | Shows progress toward proficiency, but with errors in grammar, sentence structure, and spelling, negatively impacting readability | Submission has critical errors in grammar, sentence structure, and spelling, preventing understanding of ideas | 5 |
| **Total:** | | | | | 100% |