Rukhsar Amin

6-2 Project One

Professor **Nathan Lebel**

June 15th, 2025

**My Course Object Structure:**

This project represents the integration of all the efforts I have put into this course. It provided me with an opportunity to use my ideas, structure them in a meaningful way, and think critically about the best solutions—not only from a technical standpoint but also from practical and personal perspectives. Below is my comprehensive and final submission, which includes revised pseudocodes for all milestones, a carefully designed menu layout, and an analysis of runtime presented in a manner that I found most logical.

**1: Vector**

- Open file (courses.txt)  
- Read file line by line  
- For each line:  
   - Parse course ID, name, and prerequisites  
   - Create a Course object  
   - Push Course object to a vector  
- End file

**2: Hash Table**

- Open file (courses.txt)  
- Read each line  
- Extract course ID, name, and prerequisites  
- Create Course object  
- Insert into hash table using course ID as key  
- End file

**3: Binary Search Tree (BST)**

- Open file (courses.txt)  
- Read file line by line  
- Parse and create Course object  
- Insert course into BST using course ID for sorting  
- Traverse tree in order when displaying courses

* Menu Options Pseudocode

- Display menu:  
   1. Load Data Structure  
   2. Print Course List (A-Z)  
   3. Print Course (with prerequisites)  
   4. Exit  
- Ask user for input  
- Run selected option

* Course List Display Pseudocode

- If vector or hash table:  
   - Sort the structure alphabetically by course ID  
   - Print each course ID and name  
- If BST:  
   - Use in-order traversal to print sorted courses

* **Runtime Analysis & Evaluation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Structure | Operation | Cost per line (Big-O) | Lines Executed | Total Cost |
| Vector | Search Course | O(n) | n | O(n) |
| Hash Table | Search Course | O(1) | 1 | O(1) |
| BST | Search Course | O(log n) | log n | O(log n) |

I think the hash table is the best choice for the final implementation. It is efficient, straightforward, and particularly effective when it comes to finding specific courses. However, I appreciated the organization and balance offered by the BST. While vectors are easy to use, they are not the best for scalability. Ultimately, based on my approach, I would choose the hash table it seems the most intuitive to me.