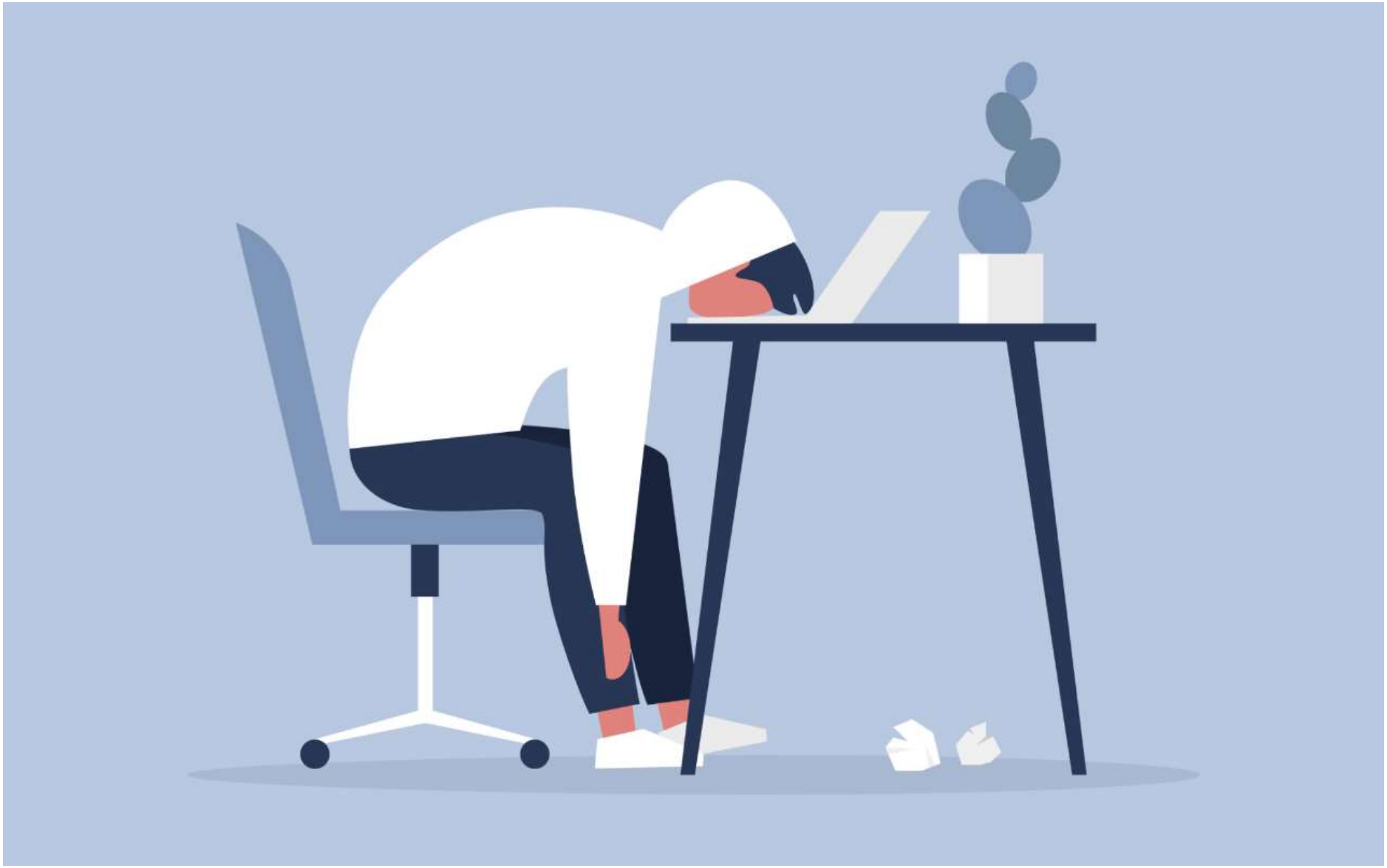
Week 7

Posted Apr 14, 2025 9:59 AM



Welcome to week seven, everyone. From here, you can see the end of the rainbow because there are only 14 days left in our time together. I'm so glad that you've made it this far and I'm sure that you now have some coding under your belt.

This week is the project2 week, where you shall begin by reading the rubrics in detail: (at least three times)

https://learn.snhu.edu/d2l/le/content/1115992/viewContent/19033960/View

What to submit:
Submit all of your C++ code that is needed to implement the project in a single ZIP file. Make sure the code compiles and runs.

I am using cmake or makefile, make sure you include them if you have more than just one single .cpp and .hpp file.

otherwise just the .cpp and .hpp file will be enough.

==========

Portfolio Prep

- It is important that you set up your GitHub repository in this module and add me as a collaborator (c.ling@snhu.edu). To this end, a non-graded activity is set up in which you are required to copy the link to your repository invite. This is an essential step for me to be able to view your portfolio in Module Eight.

==== Bonus Question ====

I'll give extra credit to those who want to boost their grade in this course. Instead of printing out the course schedule in alphanumerical order, think about how you could organize your data structure and algorithm to print the course schedule in **topological order**.

If it's not possible to complete all courses due to the prerequisite constraints, return an **empty schedule**.

Bonus: Course Schedule in Topological Order

You are given a list of courses and their prerequisites. Your task is to determine a valid order in which to take all the courses such that all prerequisites are satisfied.

If there are multiple valid topological orderings, you may return any of them. If it is impossible to complete all the courses due to a cycle in prerequisites, return an empty list.

Function Signature:

```
vector<int> findCourseSchedule(int numCourses, vector<vector<int>>& prerequisites) {}
```

Input:

- `numCourses` (int): Total number of courses labeled from `0` to `numCourses - 1`.
- `prerequisites` (vector<vector<int>>): Each element is a pair `[a, b]` meaning course a requires course b as a prerequisite.

Output:

- A list of course indices in a valid topological order.
- If no valid order exists, return an empty list.

Example 1:

```
Input: numCourses = 4, prerequisites = {{1, 0}, {2, 0}, {3, 1}, {3, 2}}
Output: [0, 1, 2, 3] // or [0, 2, 1, 3]
```

Example 2:

```
Input: numCourses = 2, prerequisites = {{1, 0}, {0, 1}}
Output: []
```

Constraints:

- `1 <= numCourses <= 10^5`
- `0 <= prerequisites.size() <= 10^5`
- All prerequisite pairs are unique and valid (i.e., 0 ≤ a, b < numCourses)

Testing Code:

```
#include <iostream>

int main() {
    int numCourses = 4;
    vector<vector<int>> prerequisites = {{1, 0}, {2, 0}, {3, 1}, {3, 2}};

    vector<int> schedule = findCourseSchedule(numCourses, prerequisites);

    for (int course : schedule) {
        cout << course << " ";
    }
    cout << endl;

    return 0;
}
```