

BCS 370 Spring 2023

Final Project Description

Project Title: College Student Homework Management System

Business Requirements: The system manages and displays a college student's homework/assignments in the current semester. The objective of the application is to help the student efficiently manage his time in doing his course work.

(To minimize your efforts, I removed the class scheduling part from the system.)

Technical Specifications:

General functional requirements:

The homework management and lookup system should provide, at a minimum, "search", "add", "update", "remove" and "display" functionalities. The "remove" function can be replaced with marking the assignment as "Completed" if you think it makes more sense.

You may use any programming language of your preference for this project.

Detailed function requirements:

1. **Data structure:** You may choose to use any data structure(s)/container(s), but **array**, you see fit for this project, STL is encouraged to use. Or you may construct a custom data structure to your needs. (2 points)
2. Must develop the project using Object Oriented Programming (OOP) (1 point)
3. Design necessary attributes for the homework/assignment object, such as name, course, due date/time, etc. (2 point)

You must include the following attributes:

- 3.1. Estimated time to spend on the assignment
- 3.2. Weight (how many credits, or percentage of the total grade)

4. Must have the following functionalities: (16 points)
 - 4.1. **Add** an assignment (2 point)
 - 4.2. **Update** an assignment (2 point) //you may update the due date, or other attributes
 - 4.3. **Search**/lookup (6 points)
 - 4.3.1. Search assignments which are due in/within # of days from today //0 means today; 1 means tomorrow; ...
 - 4.3.2. Search by course
 - 4.3.3. Display all assignments that are in progress
 - 4.4. **Delete** a completed assignment or mark the assignment as "complete" (2 points)
 - 4.5. **Save** assignments to file (binary save is the most efficient way) (2 points)
 - 4.6. **Extra Credits (3 points)** *Read assignments from file (read the assignment from file and load the data to the data structure you choose for the project.*
 - 4.7. At least one method should be a **recursive** method/function (2 points)
5. Clean and efficient code

General guideline:

Data serialization: It's recommended that you save all homework data into a file and retrieve data from the file to make the application more useful. Since each assignment is an object, [saving](#) the collection of assignment objects into a [binary file](#) will be the most efficient solution.

Modularization: Classes/Objects, methods, functions should be well thought and designed before any implementation taken place. (Of course, the design is likely to be updated along with the development.) A [UML diagram](#) will be required as part of the project submission.

Submission Requirements:

You may create a compressed file of all the following files to submit.

1. Team writeup (7 points)/Individual Team (10 points) [2 pages]
 - a. Team member(s)
 - b. [UML](#) diagram
 - c. Key technical implementation descriptions and explanation
 - i. data structure
 - ii. recursive function
 - iii. complexity of `searchByDaysFromDueDate()` and `searchByCourse()`
 - d. References if any

2. Source code (GitHub link is okay)

3. If Teamwork, individual writeup (3 points) [1 page]

Each member should write up a one-page document to describe his contribution to the project. (must include technical contribution)

- a. Design portion (which part in the UML, data structure) if applicable.
- b. Coding portion if applicable.
- c. Describe and explain your contribution.