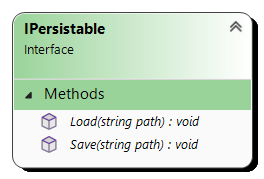
# Assignment 6 and 7: Exceptions and Interfaces in C# - School Task Management System

Objective:

# The objective of this assignment is to extend the existing School Task Management System to include new features and improvements, promoting a more robust and versatile system. The modifications focus on introducing an IPersistable interface, refining existing classes, and handling exceptions.

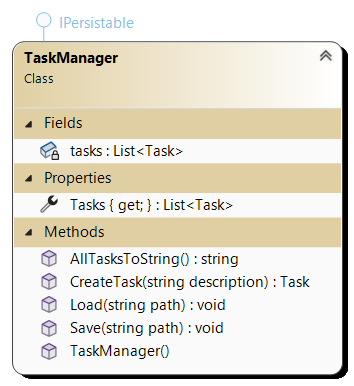
# **Please observe the General assignment requirements outlined in the document on e-centennial**

## IPersistable Interface



[10] Create interface as per class diagram. Implement all and only the members in the class diagram.

## TaskManager Class

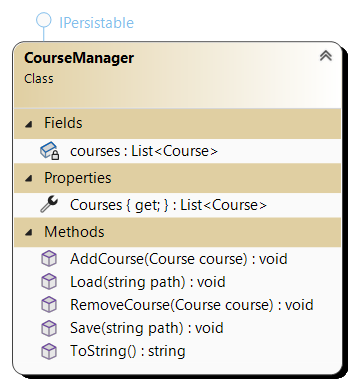


[10] Modify a TaskManager class that is responsible for **managing all** tasks. Implement all and only the members in the class diagram. Note that class is not static. None of the methods are static either.

[10] The default constructor initialises the list.

[30] Implement IPesistable interface. Make sure that you handle exception in both methods implementations. The responsibility and signatures of Load and Save methods have not changed.

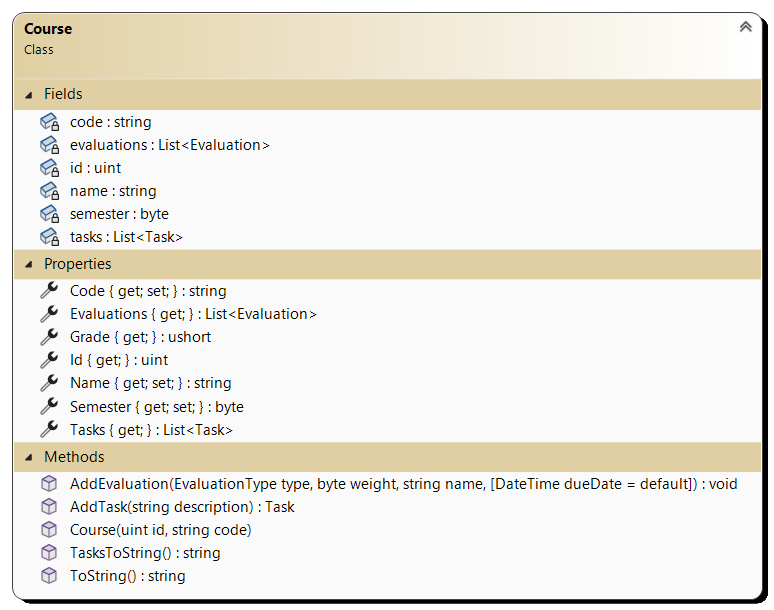
## CourseManager Class



[10] Modify a ClassManager class responsible for managing courses. Implement all and only the members in the class diagram. The class is not static. None of the methods are static either.

[30] Implement IPesistable interface. Make sure that you handle exception in both methods implementations. The responsibility and signatures of Load and Save methods have not changed.

## Course Class



[10] Modify a Course class that represent a course. Implement all and only the members in the class diagram.

Modify method AddEvaluation(EvaluationType type, byte weight, string name, DateTime dueDate = default) to throw an exception if due date or weight are not valid.

a. [10] When provided due date is in the past the ArgumentException is thrown with message: "Due date must be in the future."

b. [20] If the total weight of all evaluation would exceed 100% when the new evaluation is added the ArgumentException is thrown with message: "Total evaluations weight exceeds 100%".

Other members are not changed.

## Evaluation Class

A screenshot of a computer

Description automatically generated

[10] Modify an Evaluation class that represent an evaluation within a course. Implement all and only the members in the class diagram.

[15] Modify Weight property to throw AgumentException with the message: "Total evaluations weight exceeds 100%" if the attempt is made to set weight for an evaluation that would cause total weight for all evaluations to exceed 100%.

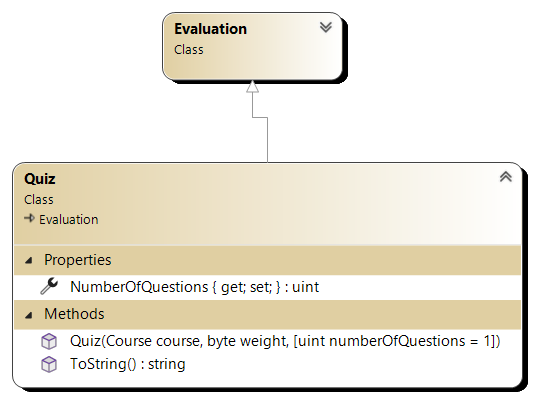
[10] Modify property EvaluationText that return the text in the file specified by textFile field to include exception handling.

## Assignment Class

## 

The Assignment class is not changed.

## Quiz Class



The Quiz class is not changed.

## MyDay Class

A screenshot of a computer

Description automatically generated

This type has not changed.

## Task Class:

A screenshot of a task

Description automatically generated

The Assignment class is not changed.

The Task class is not changed.

## EvaluationType enumeration

A screenshot of a computer

Description automatically generated

This type has not changed.

## Test harness

[100] Demonstrate the functionality of the Task Management System in the Main method by doing the following:

1. Create instance of both TaskManager and CourseManager classes
2. Create 2 courses
3. Add courses to the course manager
4. Write content of course manager to the console
5. Add 2 assignments to the first course with due dates in 14 and 7 days respectively
6. Add quiz, and test to the first course
7. Add test to the first course
8. Add 3 assignments to the second course with due dates in 10, 5, and 6 days respectively
9. Trying to add assignment with a due date in the past
10. Trying to add assignment with a weight that exceeds total weight of 100% for all assignments
11. Trying to modify the weight of an assignment to exceed total weight of 100% for all assignments
12. Add test to the first course
13. Set TextFile for the first assignment of the first course to 'Evaluations\COMP123\Assignmnet1.txt'
14. Write content of the evaluation text for the first assignment of the first course
15. Write content of course manager to the console
16. Add grade to the first assignment of the first course
17. Write content of course manager to the console
18. Add grade to the second assignment of the first course
19. Write content of course manager to the console
20. Add task 'Read chapter 2' to the first course with due date in 6 days
21. Add task 'Read chapter 3' to the first course
22. Output tasks for the first course to the console
23. Add task 'Start assignment' to the first assignment of the first course
24. Add task 'Write main() method' to the first assignment of the first course and mark it done
25. Output all task for the first evaluation of the first course to the console
26. Create MyDay and add 2 tasks
27. Output the MyDay to the console
28. Create array of 2 IPersistable elements and add instances of class and task manager
29. Persist all elements of the array using Save method

The file is generated in the Save folder that is the subfolder of the folder containing the executable file. The name of the file is <type>.json where <type> is the name of the type that is persisted.

1. Create instance of both TaskManager and CourseManager classes and add them to elements of the array
2. Load all persisted content from a file in each element of the array
3. Output a separator
4. Write content of course manager to the console
5. Write all tasks in the task manager to the console

The output for above test harness should look similar to the following one provided in output67.txt file.

## Submission Guidelines:

Submit your solution as compressed solution folder in Lab 67 drop box by the decline in the drop box.

Include a brief written explanation of design choices, challenges faced, and lessons learned during implementation. Submit the explanation as a word document. Place the word document in the solution folder together with the solution file.