

Sofia University
Department of Mathematics and Informatics

Course : **OO Programming with C#.NET**

Date: October 13, 2020

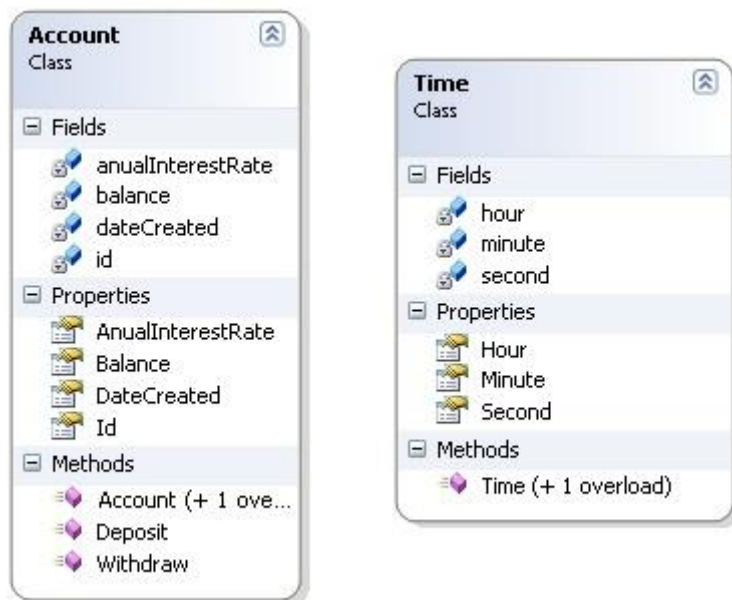
Student Name:

Lab No. 2a

Submit the all C# .NET files developed to solve the problems listed below. Use comments and Modified-Hungarian notation.

Problem No. 1

Use VS Visual designer to produce the following classes



Problem No. 2

Modify *class GradeBook* from **Fig. 4.12** as follows:

- a) Include a second `string` instance variable that represents the name of the course's instructor.
- b) Introduce a getter property `CourseStart` to hold the current year of the `GradeBook`
- c) Provide a property with accessors to change the instructor's name and to retrieve it employing an auto-implemented property.
- d) Modify the constructor to specify two parameters one for the course name and one for the instructor's name.

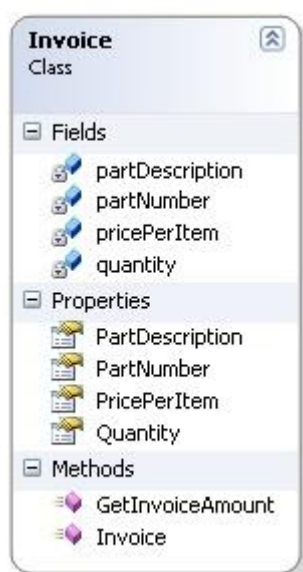
- e) Modify method `DisplayMessage` such that it first outputs the welcome message and course name, then outputs "This course is presented by: ", followed by the instructor's name. Use String interpolation.
- f) Add a public method `GradeBookTitle` returning a tuple with the values of the `CourseStart`, `CourseName` and `instructor`
- g) Add a public method `ChangeCourseTitle` that accepts a tuple named `title` with two strings for instructor name and course name. The method updates the tuple made of `CourseName` and `instructor` with the values passed by `title`

Use your modified class in a test application that demonstrates the class's new capabilities.

Problem No. 3

Employing the Visual Designer develop a model for a class called *Invoice* that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four pieces of information as instance variables - a part number (type *string*), a part description (type *string*), a quantity of the item being purchased (type *int*) and a price per item (*decimal*). Your class should have a *constructor* that initializes the four instance variables. Provide a *property* with a *get* and *set* accessor for each instance variable. Generate the necessary C# code and in addition provide a method named *GetInvoiceAmount* that calculates the invoice *amount* (i.e., multiplies the quantity by the price per item), then returns the amount as a decimal value. If the *quantity* is negative, it should be left unchanged. Similarly, if the *price* per item is negative, it should be left unchanged.

Write a test application named *InvoiceTest* that demonstrates class Invoice's capabilities.



Problem No. 4

What does the following application print?

```
1 // Ex. 5.25: Mystery2.cs
2 using System;
3
4 public class Mystery2
5 {
6     public static void Main( string[] args )
7     {
8         int count = 1;
9
10        while ( count <= 10 )
11        {
12            Console.WriteLine( count % 2 == 1 ? "****" : "+++++++" );
13            count++;
14        } // end while
15    } // end Main
16 } // end class Mystery2
```

Problem No. 5

Identify and correct the errors in each of the following pieces of code. [Note: There may be more than one error in each piece of code.]

- a.
- ```
 if (age >= 65);
 Console.WriteLine("Age greater than or equal to 65");
 else
 Console.WriteLine("Age is less than 65");
```
- b.
- ```
    int x = 1, total;
    while ( x <= 10 )
    {
        total += x;
        ++x;
    }
```
- c.
- ```
 while (x <= 100)
 total += x;
 ++x;
```
- d.
- ```
    while ( y > 0 )
    {
        Console.WriteLine( y );
        ++y;
```