Lab 2

# Objectives

Practice C# code writing that involves Objects and Classes other object oriented programming concepts.

# Instructions

* Install Visual Studio 2017 or Visual Studio 2019 to your computer.<https://visualstudio.microsoft.com/>
* For each question below submit your C # and also screenshots showing how your program compiles and executes (sample outputs)- You can upload your C# project but alternatively create a document with your C# code **text** copied and pasted and your screenshots pasted and then upload the document to Blackboard for submission. Prof. Aydin should be able to run your submitted code.
* Your name should appear on the screenshots for receiving full credit.
* **Academic Integrity:** If you are stuck when working in this lab you can collaborate with a couple of classmates. In that case, make sure to write/submit the name of your collaborators and any web site you used as a resource to understand the concepts and lab questions and to complete your code to prevent plagiarism and breach of academic integrity.
  + You are *not* allowed to directly copy code from the Internet, your friends, and other resources without spending any effort in completing the work. Make sure to review the academic integrity policy in the syllabus and ask for clarification, if needed.

1. Create a C# class called Document
   1. with the following Properties:
      1. Name (string)
      2. NumberOfPages (int)
      3. CreatedOn (DateTime): make this a **readonly** property. Note that [DateTime](https://docs.microsoft.com/en-us/dotnet/api/system.datetime?view=netcore-3.1) is a struct in the namespace System
   2. Add a constructor to allow initialization of all the properties ~~setting properties Name and NumberOfPages front~~
   3. Add ToString Method (hint: override toString method of [System.Object](https://docs.microsoft.com/en-us/dotnet/api/system.object?view=netcore-3.1) class)

Then, write a Main method to test your class Document.

1. Create document1 with name “PhoneNumbers.txt” and 3 pages
2. Create document2 with name “Report.docx” and 30 pages
3. Demo how you cannot change the CreatedOn property of each document object

2. Create a class Employee

a) *private* and *static field* nextAvailableID (int): create a *static constructor* to initialize this field to a random number (hint: you can use [System.Random class](https://docs.microsoft.com/en-us/dotnet/api/system.random?view=netcore-3.1)). You can assume that IDs are numbers between 1-9999. Since nextAvailableID is static, all the Employee objects will share this field.

b) with properties

* FirstName (string)
* LastName (string),
* ID (int): make it **read only** to allow initializing only with a constructor and then no more.

c) Add a constructor to allow setting properties FirstName and LastName. Hint: ID will be initialized to nextAvailableID (make sure to increment nextAvailableID by one)

1. Add ToString Method to display the full name and ID of the employee

Then, write a Main method to test your class Employee.

1. Create four Employee objects with references e1, e2, and e3 by providing first and last name of each employee
2. Display the full name and id of each employee - note that each employee will have an automatically generated ID by the class Employee.

A sample output of your program the first time it runs could be as below.



Next time, the program runs, employees will be assigned different but again continuing IDs.

