C# Basics – Day 3 and 4 – Homework

Write a program that creates user accounts that are accessible through the lifetime of a single application run (when it is restarted - list is always empty). On the start of application, we should store a date and time when application was started in some place that is available everywhere in the application without instantiating any class.

Then application should display a menu:

```
I.   Create new user
II.  Display Users
III. Delete user
IV.  Exit program
```

More specification:

I. Create new user

- Single user should have following properties: First Name, Last Name, Age, Email, Password, IsActive

- Email cannot be set explicitly - it can be only read outside of the class and is always created as first two letters of first name and two letters of last name at example.com domain. For instance: John Smith should have email set to josm@example.com (You can use a public method GenerateEmail() without parameters that uses previously set First and Last name)

- as we didn't speak about instantiating object from constructor with parameters – using this approach could grant you extra points. Example for Car class:

```
public Car(string brand, int maxSpeed) {
     MaxSpeed= maxSpeed;
     Brand = Brand;

      // analogy for auto email
     someAutomatedProperty = CalculatePoperty(Brand, MaxSpeed);

}
```

And then:

```
var car = new Car("Volvo", 220);
```

There is also a way to create a new object with public properties:

```
var brand = "Volvo";
var maxSpeed = 220;

var car = new Car {Brand = brand, MaxSpeed = maxSpeed};
```

- In addition: we only allow users that are between 18 and 65 years old, so whenever we try to create a user below 18 or above 65, it should be impossible to set it - application should throw the

exception of type InvalidArgumentException with explanation what field is it and why it is not correct - this should abort adding a user and go back to the main menu with correct message.

(some reference: https://docs.microsoft.com/pl-pl/dotnet/standard/design-guidelines/using-standard-exception-types)

- Password should be a random string of length between 16 and 32. It cannot be expicitly set but should be generated by calling public GeneratePassword() method in User class - this method inside should call a static helper method outside of User class (for setting random length use "Random" class or try to find some "random string generator" on stack overflow :). Also Path.GetRandomFileName() can be somehow used).

- when user email is already taken - a number should be added at the end of username, ie:

1. John Monroe => jomo@example.com
2. Josh Moon => jomo1@example.com
3. Johannes Moulder => jomo2@example.com

II. Display users

There should be information about number of current users and they should be displayed in a form of a list, i.e.:

#	Name	Last Name	Email	Age	IsActive	Password
1.	John	Smith	josm@example.com	43	true	ks87djHJjdj87sdf
2.	Jane	Doe	jado@example.com	21	false	j76shd76s8JH8d7s

III. Delete user

- to delete the user: we must enter his email, if user not found - throw exception: "ArgumentException" and show appropriate message.

IV. Exit program

- Write in the console: Application was running for XX minutes and YY seconds, ZZ users are present

Some implementation details and level of details in this specification is limited ON PURPOSE to give you some freedom in interpretation and reflect real life level of documentation that you can encounter 😊