Lab#05: Object Oriented Programming- COC2071

Instructor: Abdul Qadeer Bilal

Registration #		Name	
Total Marks	20 marks (10 marks each)	Marks Obtained	
Tools	Visual Studio		
Objectives	1) Static Classes	2) Inheritance	
Note	Solve the following problems using the concepts we have covered so far		

QUESTION:1

You are tasked with developing a simple employee management system. Create the following classes and a static class for basic features:

Employee Class:

- Properties: EmployeeID, Name, and Salary.
- Static Variable: Create a static variable **nextEmployeeID** that keeps track of the next available employee ID. Initialize it in the static constructor.
- Static Method: **GenerateEmployeeID()** to generate the next employee ID.

Methods:

DisplayEmployeeInfo() to display the employee information.

GiveRaise(double amount) to increase the employee's salary.

Department Class:

Static Properties: DepartmentID and Name and a list (array) of Employees in the Department.

Static Methods:

DisplayDepartmentInfo() to display the department information and all employees in the Department.

EmployeeUtilities Static Class:

Static Method:

AssignEmployeeToDepartment(Employee employee) to assign an employee to a department.

CalculateAnnualSalary(Employee employee) to calculate and return the annual salary of an employee (assuming 12 months of pay).

Your task is to create instances of the **Employee** and **Department** classes, use the static constructors to manage IDs, and simulate the following scenario:

1. Create at least three employees with different names and salaries.
2. Assign employees to department.
3. Give raises to some employees.
4. Display employees information with in department using the static methods in the **Department Class.**

QUESTION:2

You are tasked with creating a simple social network system.

User Class:

Create a base class called **User** to represent a social network user.

Properties:

- **UserID** (a unique identifier for each user).
- **Username** (the user's username).
- **Posts** (a list to store the user's posts).
- ~~**Comments** (a list to store the user's comments on other users' posts).~~

Methods:

- **AddPost(string postContent)** to add a post to the user's profile.
- ~~**AddComment(Post post, string commentContent)** to add a comment to a post.~~
- **DisplayProfile()** to display the user's profile, including their username, posts, ~~and comments.~~

Admin Class:

Create a derived class called **Admin** that inherits from the **User** class.

- Additional Property:

Permissions (a string of permissions assigned to the admin). It would be like Edit, Add, Delete etc.

- Methods:
AddPermission(string permission) to add a permission to the admin's profile.
Override **DisplayProfile()** to display the admin's profile, including their username, posts, ~~comments~~, and permissions.

Moderator Class:

Create a derived class called **Moderator** that inherits from the **User** class.

- Additional Property:

Permissions (a string of permissions assigned to the Moderator). It would be like Edit Only etc.

- Methods:
  AddPermission(string permission) to add a permission to the Moderator's profile.
  Override **DisplayProfile()** to display the Moderator's profile, including their username, posts, ~~comments~~, and permissions.

Your task is to create instances of the **Moderator** and **Admin** classes, demonstrate inheritance, and perform the following operations:

1. Create at least two **Moderator** objects with different usernames and add permissions to it.
2. Create at least one **Admin** object with a unique username and add permissions to it.
3. Have the Moderator and admin create posts ~~and add comments~~.
4. Display the profiles of all Moderators and the admin, including their posts, ~~comments~~, and permissions (for the admin).