National Textile University

Department of Computer Science

Lab#10: Object Oriented Programming- COC2071

Instructor: Abdul Qadeer Bilal

Registration #		Name	
Total Marks	10 marks	Marks Obtained	
Tools	Visual Studio		
Objectives	1) Windows Forms		
Note	Solve the following problems using the concepts we have covered so far		

QUESTION:1

You have been tasked with developing a Student Management System using Windows Forms in C#. The system should allow users to perform basic CRUD operations on student records. Additionally, the solution must be organized using Object-Oriented Programming (OOP) principles, utilizing classes and interfaces.

Requirements:

User Interface:

- Create a Windows Form application with the following components:
- A DataGridView to display a list of student records.
- Textboxes for entering student information (e.g., ID, Name, Age, and Grade).
- Buttons for CRUD operations: "Add," "Update," "Delete," and "Refresh."
- The form should have a clean and intuitive layout. Use labels to identify each textbox, and ensure the DataGridView is well-organized.

OOP Implementation:

- Define a class named `Student` with properties for ID, Name, Age, and Grade.
- Implement an interface named `IStudentRepository` with methods for CRUD operations: `AddStudent`, `GetAllStudents`, `UpdateStudent`, and `DeleteStudent`.
- Create a class `StudentRepository` that implements the `IStudentRepository` interface. This class should use a List to store student records.
- Ensure that the Windows Form interacts with the `StudentRepository` class to perform CRUD operations on student records.

Database Integration:

- Use a SQL Server Database to store student records.
- Modify the `StudentRepository` class to interact with the database for CRUD operations.
- Provide clear instructions in your repository on how to set up and configure the database connection.

Student Management System Interface (Windows Forms):

Main Window:

- DataGridView: Displaying a list of student records in a tabular format. Each row represents a student with columns for ID, Name, Age, and Grade.
- Add Button: Initiates the process of adding a new student. Clicking this button should clear the textboxes for entering new student information.
- Update Button: Enables the user to update the selected student's information based on the entered values in the textboxes.
- Delete Button: Deletes the selected student from the system.
- Refresh Button: Reloads and updates the DataGridView with the latest student records.

Textbox Section:

- ID Textbox: Allowing the user to enter the ID of the student.
- Name Textbox: For entering the student's name.
- Age Textbox: Capturing the student's age.
- Grade Textbox: Accepting the student's grade.

Labeling:

- Labels should be placed next to each textbox to clearly indicate the purpose of each input field.

Layout and Design Suggestions:

- Keep the layout clean and organized, with a logical arrangement of components.
- Use appropriate spacing to avoid a cluttered appearance.
- Group related components together to enhance usability.
- Consider using a consistent and visually appealing color scheme.

Mockup:

| Student Management System |

[DataGridView – Layout – Here 1st row is header and rest are Data]

| ID | Name | Age | Grade |

| 1 | AQ | 20 | A |

| 2 | Umair | 22 | B |

|…|…|…|…|

[Add] [Update] [Delete] [Refresh]

ID: [ ]

Name: [ ]

Age: [ ]

Grade: [ ]