



**National University of Computer & Emerging
Sciences**

Islamabad Campus

Object Oriented Programming (OOP)

PROJECT REPORT

Submitted By: Urwah Ejaz

Roll Number: 22i-0474

Submission Date: May, 12, 2023

Class-Section: Ai-B

Instructor: Sir Adil Majeed

Department of Artificial Intelligence



Table of Contents

1.	Introduction	3
2.	Overview of the System	3
3.	OOP Concepts	4
3.1.	Inheritance	4
3.2.	Polymorphism	4
3.3.	Aggregation	5
3.4.	Composition	5
3.5.	Const / Static Data	6
4.	System Screenshots	6
4.1.	Main menu	6
4.2.	Login Page	7
4.3.	Admin Panel	7
4.4.	Teacher Panel	8
4.5.	Student Panel	8
5.	System Usage Guide	9
6.	Class Diagram	10

1. Introduction

This report discusses how an examination system was designed and implemented using Object-Oriented Programming (OOP) principles. The system creates data and analytics regarding student performance and enables a teacher to design and manage course assessments, including quizzes and assignments. Teachers and students are the two different user categories for the console-based system.

The study covers how the system was implemented using OOP principles including inheritance, aggregation, association, composition, and constant/static data members. It also contains a class diagram that shows the scenario and screenshots to accompany the description. The paper also offers instructions on how to use the system's many capabilities.

2. Overview of the System

The objective of the examination system project is to develop and construct a console-based application that will enable the teacher to manage the course evaluations by putting up quizzes and assignments, creating a question bank, administering and grading evaluations, and producing analytics reports. The system will use file reading and writing to permanently save data on the disc.

The instructor and the student will be the two different user types for the system. The instructor will be in charge of building a question bank, producing tests and assignments, and deciding when to hold assessments. The system will automatically grade the exams, provide grades and analytics data, and the students will try the exams at the designated time.

The following data flows will be implemented as part of the project: user creation, course offering, course registration, preparation of the question bank, quiz setup and generation, conduct, generation of the answer key, preparation of the result, and quiz analytics.

In order to accomplish the project, Object-Oriented Programming (OOP) techniques including inheritance, aggregation, association, composition, and constant/static data members will be employed to describe the problem. The project will include the necessary menus to request the user's input as needed, deal with any incorrect inputs, and handle exceptions if they arise. There should never be a program crash.

Additionally, a method for authentication will be introduced, requiring users to login and create a password. When users input a password, the system will show asterisks (*), and users will have the ability to modify the password through the relevant menu. The password must be at least six characters long and include uppercase, lowercase, special characters, and digits.

3. OOP Concepts

3.1. Inheritance

Inheritance has been used at multiple places in the implementation of this project. This includes

- Student, Teacher and Admin class inherited from User Class.
- Assignment & Quiz class inherited from Assessment Items Class.
- MCQ, Descriptive and TrueFalse classes inherited from Questions class.

3.2. Polymorphism

Polymorphism has been used when MCQ, Descriptive and TrueFalse classes inherited from Questions class. pure virtual functions are made.

e.g.

```
class Question {
    string topic;
    string type;
    int wrong;
    int right;
public:
    Question() { ... }
    Question(string type) { ... }
    Question(string topic, string type) { ... }
    void virtual createQuestion(string topic) = 0;
    int virtual print() = 0;
    void virtual writeToFile(ofstream& file) = 0;
    void setTopic(string topic) { ... }
    void setType(string type) { ... }
    string getType() { return type; }
    string getTopic() { return topic; }
    int getWrong() { return wrong; }
    int getRight() { return right; }
    void addWrong() { wrong += 1; }
    void addRight() { right += 1; }
    string virtual getQuestion() = 0;
};
```

3.3. Aggregation

Aggregation is implemented between student and course, teacher & course.

```
#include "Topic.h"
#include <fstream>
#include "AssessmentItems.h"
#include "Result.h"
class Student;
class Teacher;
class Course {
    string courseName;
    string courseCode;
    Topic** topic;
    int topicsCreated;
    int maxTopics;
    Teacher* instructor;
    Student** students;
    AssessmentItems** assessments;
```

```
class Student : public User {
    Course** courses;
    int* courseCount;
    int* courseMax;
    Student** students;
```

```
class Teacher : public User {
    Course** courses;
    int* courseCount;
    int* courseMax;
public:
```

3.4. Composition

Composition is between assessment items and date time, assessment items & attendance, topic and question bank.

```
class AssessmentItems {
    static int count;
    int id;
    DateTime date;
    AssessmentAttendance* attendance;
    Topic* topic;
```

```
class Topic {
    QuestionBank* questionBank;
    string topic;
public:
```

3.5. Const / Static Data

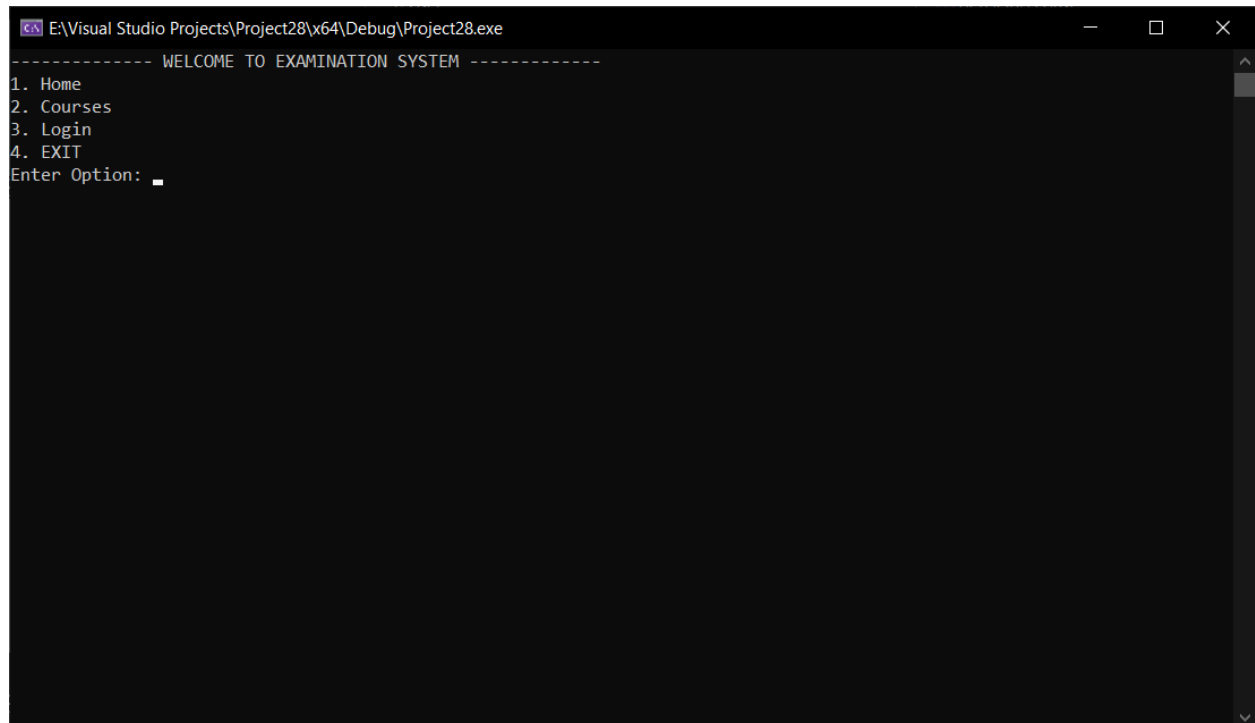
In assessment items, the count variable is made to make a unique id by the auto increment method. It keeps track of a number of created classes.

```
class AssessmentItems {
    static int count;
    int id;
    DateTime date;
    AssessmentAttendance* attendance;

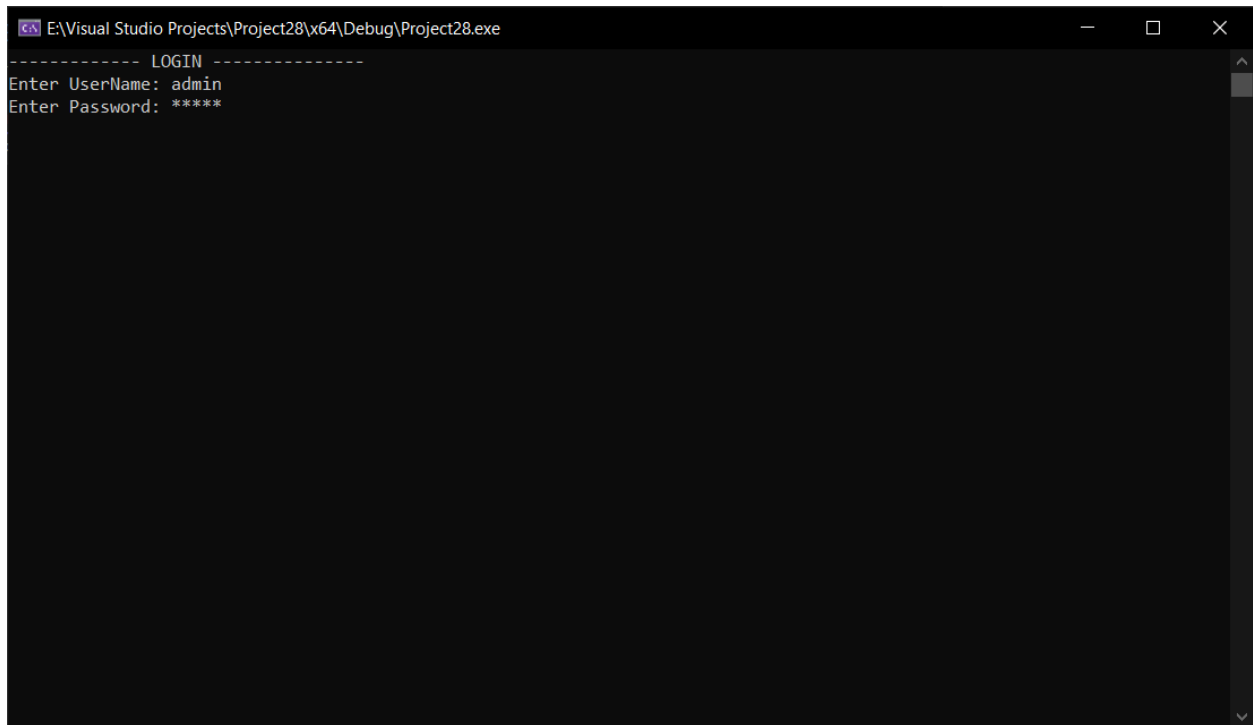
    const string& getName() const { return name; }
    const string& getAddress() const { return address; }
    const string& getEmail() const { return email; }
    const string& getDOB() const { return dob; }
    const string& getUsername() const { return username; }
```

4. System Screenshots

4.1. Main menu

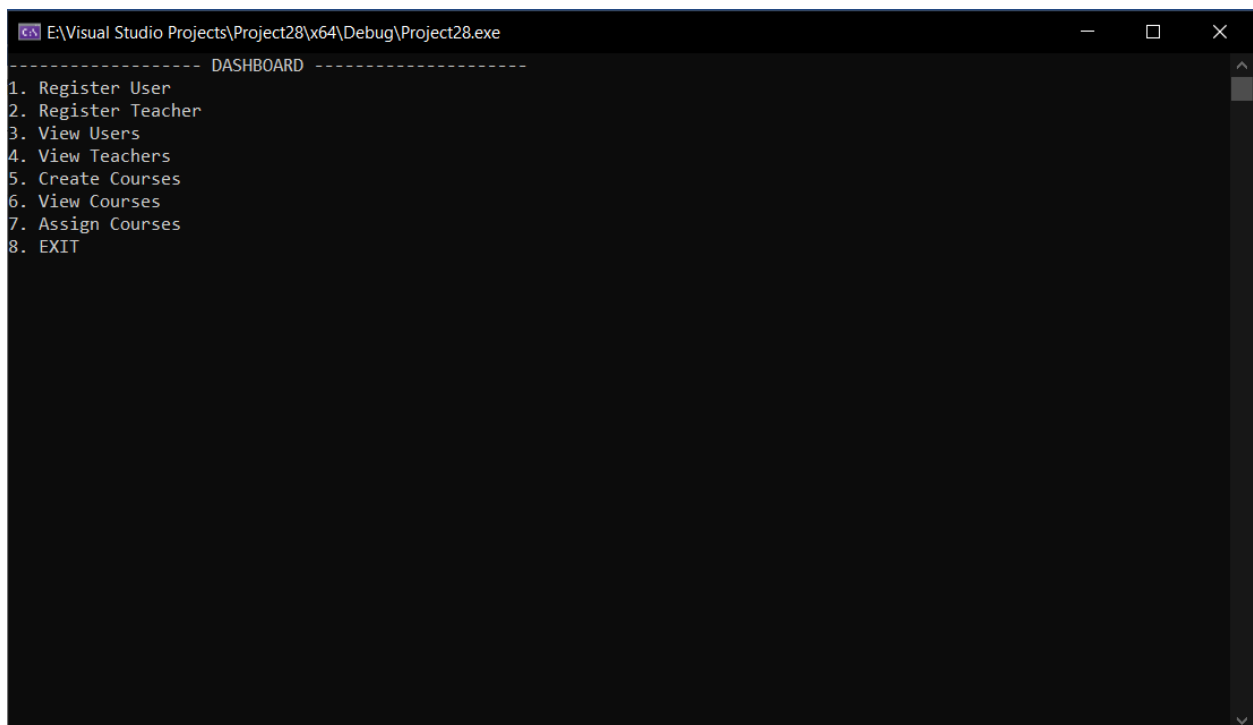


4.2. Login Page



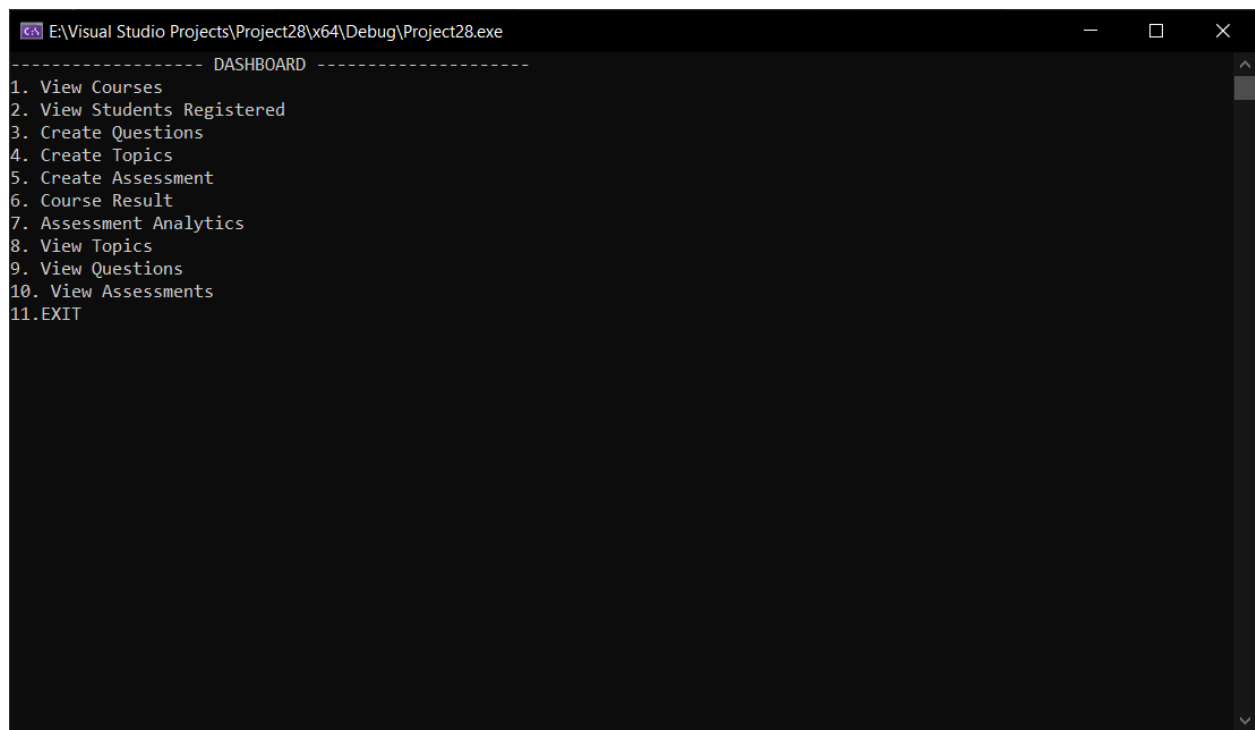
```
E:\Visual Studio Projects\Project28\x64\Debug\Project28.exe
----- LOGIN -----
Enter UserName: admin
Enter Password: *****
```

4.3. Admin Panel



```
E:\Visual Studio Projects\Project28\x64\Debug\Project28.exe
----- DASHBOARD -----
1. Register User
2. Register Teacher
3. View Users
4. View Teachers
5. Create Courses
6. View Courses
7. Assign Courses
8. EXIT
```

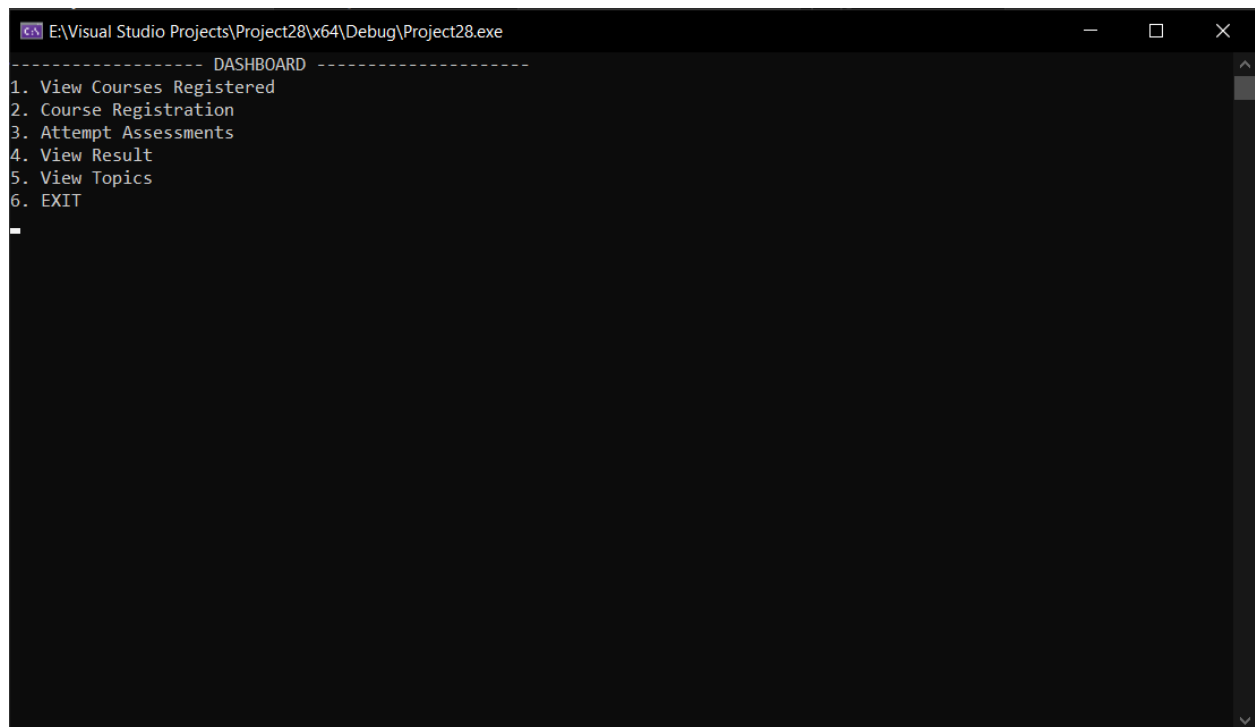
4.4. Teacher Panel



The screenshot shows a Visual Studio window with the file path `E:\Visual Studio Projects\Project28\x64\Debug\Project28.exe`. The application displays a dashboard with a list of 11 menu items. The text is as follows:

```
----- DASHBOARD -----
1. View Courses
2. View Students Registered
3. Create Questions
4. Create Topics
5. Create Assessment
6. Course Result
7. Assessment Analytics
8. View Topics
9. View Questions
10. View Assessments
11.EXIT
```

4.5. Student Panel



The screenshot shows a Visual Studio window with the file path `E:\Visual Studio Projects\Project28\x64\Debug\Project28.exe`. The application displays a dashboard with a list of 6 menu items. The text is as follows:

```
----- DASHBOARD -----
1. View Courses Registered
2. Course Registration
3. Attempt Assessments
4. View Result
5. View Topics
6. EXIT
```


5. System Usage Guide

At the start, you will be seeing a data loading page. Just press any key to continue. Now head over to the login page and enter your credentials if you are registered. Otherwise, you will get basic info from the Courses and Home Page. After Login enter option number to continue...

6. Class Diagram

