



# Uttara University

Excellence in Higher Education and Research

**Department of**  
**COMPUTER SCIENCE AND ENGINEERING**  
**PROJECT REPORT**

**Course Code: CSE0613202**

**Course Name: Object-Oriented Programming Language Lab**

**Submitted By**

Nane & Id : Rashedul Islam Rifat– 2243081072

Nane & Id : Sanjida Alam Moonmoon–2243081071

Nane & Id : Fardin Fahad – 2243081172

Nane & Id : Amadul Haque Siam – 2243081050

Nane & Id : Md. Adnan Arefin Ratul– 2243081062

**Submitted To**

Name : Nahrin Jannat

Designation : Lecturer, Dept. of CSE

❖ **Project Name**-Student Result and Attendance Management System (C#)

❖ **Project Objective-**

1. To build a console-based system to manage student results.
2. To apply Object-Oriented Programming concepts using C#.
3. To calculate CGPA from marks automatically.
4. To manage attendance and generate reports for students.

❖ **Review and Limitations-**

1. Supports only OOP, DLD, and CA subjects.
2. Attendance is stored without dates.
3. No validation for marks input.
4. CGPA grading scale is fixed inside the code.
5. Data is not stored permanently.

❖ **Member Contributions & Feature Description (Code-Based)**

**Sanjida Alam Moonmoon-2243081071**

➤ **Features Used-**

1. Abstract Class
2. Interface
3. Base Class Constructor
4. Abstract Method

➤ **Used in Code-**

1. abstract class Person
2. Public variables: Name, Id
3. Constructor: Person(string name, string id)
4. Abstract method: public abstract void Show()
5. Interface: IReport
6. Interface method: string CreateReport()

➤ **How It Was Used-**

1. The Person class acts as a base category for students.

2. The constructor initializes name and ID from the derived class.
3. The abstract method Show() forces the Student class to define its own display logic.
4. The interface IReport ensures every student can generate a report string.

➤ **Review-**

1. This structure helped separate identity, result display, and reporting logic.
2. It made the system organized and avoided writing repeated code.

**Rashidul Islam Rifat-2243081072**

➤ **Features Used-**

1. Copy Constructor
2. Encapsulation
3. Private Method

➤ **Used in Code-**

1. Class: Result
2. Public variables: OOP, DLD, CA
3. Copy constructor: Result(Result r)
4. Private method: double CGPA(double mark)
5. Public method: double AverageCGPA()

➤ **How It Was Used-**

1. The copy constructor creates a new result object when assigning marks to a student.
2. The CGPA grading logic is hidden inside a private method.
3. Only the final CGPA value is accessed through AverageCGPA().

➤ **Review-**

1. This prevented unintended modification of marks and protected grading logic.
2. Encapsulation improved data safety and code reliability.

## **Fardin Fahad – 2243081172**

### ➤ **Features Used-**

1. Parameterized Constructor
2. Method Overloading

### ➤ **Used in Code-**

1. Constructor: Result(double oop, double dld, double ca)
2. Overloaded methods:
3. AddMark(double oop, double dld, double ca)
4. AddMark(string subject, double mark)

### ➤ **How It Was Used-**

1. Marks are initialized directly when the result object is created.
2. Overloaded methods allow updating marks either all at once or subject-wise.

### ➤ **Review-**

1. This reduced redundancy and increased flexibility.
2. The same method name performing different tasks made the code easier to use.

## **Amadul Haque Siam – 2243081050**

### ➤ **Features Used-**

1. List Collection
2. Looping
3. Conditional Statements

### ➤ **Used in Code-**

1. Private list: List<bool> attendance
2. Method: MarkAttendance(bool present)
3. Method: AttendancePercentage()

4. Loop: foreach
5. Condition: if (a) presentDays++

➤ **How It Was Used-**

1. Attendance status is stored dynamically using a list.
2. Attendance percentage is calculated using loop and condition logic.

➤ **Review-**

1. Using collections allowed unlimited attendance entries.
2. Type casting was necessary to avoid incorrect percentage calculation.

**Md. Adnan Arefin Ratul– 2243081062**

➤ **Features Used-**

1. Inheritance
2. Polymorphism
3. Method Overriding

➤ **Used in Code-**

1. Class declaration: class Student : Person, IReport
2. Overridden method: public override void Show()
3. Implemented method: CreateReport()

➤ **How It Was Used-**

1. Student inherits name and ID from Person.
2. Overrides Show() to display student-specific details.
3. Uses polymorphism to call the correct Show() method at runtime.

➤ **Review-**

1. Inheritance reduced code duplication.
2. Polymorphism allowed flexible method behavior with a common interface.

## Main Program Implementation

### ➤ Used in Code-

- Class: Program
- Method: static void Main()
- Object creation using constructors

### ➤ Method calls-

- MarkAttendance()
- Show()
- CreateReport()
- Collection: List<Student>

### ➤ Review

- The Main() method connects all classes and features together.
- It demonstrates real object interaction and program execution flow.

### ➤ Output-

- Displays student name, ID, subject marks, CGPA, and attendance.
- Shows individual student information.
- Generates reports for all students at the end.

### ➤ All Reports-

```
--- All Reports ---
Student Report √ Rashedul Islam Rifat | OOP: 85, DLD: 80, CA: 78, Average CGPA: 3.92, Attendance: 66.67%
Student Report √ Sanjida Alam Moonmoon | OOP: 80, DLD: 70, CA: 85, Average CGPA: 3.83, Attendance: 100.00%
Student Report √ FARDIN FAHAD | OOP: 60, DLD: 85, CA: 74, Average CGPA: 3.50, Attendance: 66.67%
Student Report √ AMDADUL HAQUE SIAM | OOP: 65, DLD: 70, CA: 60, Average CGPA: 3.25, Attendance: 100.00%
Student Report √ MD. ADNAN AREFIN RATUL | OOP: 71, DLD: 74, CA: 80, Average CGPA: 3.67, Attendance: 66.67%
PS C:\Users\PC\Desktop\project oop>
```

## ➤ Individual Report-



```
File Edit Selection View Go Run ... ← → Q project oop [Icons]

EXPLORER
└─ PROJECT OOP
  ├── bin
  ├── obj
  ├── Program.cs
  ├── project oop.csproj
  └── project oop.sln

OUTLINE
TIMELINE
SOLUTION EXPLORER

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

=== STUDENT RESULT SYSTEM ===

--- Rashedul Islam Rifat ---
[STUDENT INFO]
Name: Rashedul Islam Rifat, ID: 2243081072
OOP: 85, DLD: 80, CA: 78, Average CGPA: 3.92
Attendance: 66.67%

--- Sanjida Alam Moonmoon ---
[STUDENT INFO]
Name: Sanjida Alam Moonmoon, ID: 2243081071
OOP: 80, DLD: 70, CA: 85, Average CGPA: 3.83
Attendance: 100.00%

--- FARDIN FAHAD ---
[STUDENT INFO]
Name: FARDIN FAHAD, ID: 2243081172
OOP: 60, DLD: 85, CA: 74, Average CGPA: 3.50
Attendance: 66.67%

--- AMDADUL HAQUE SIAM ---
[STUDENT INFO]
Name: AMDADUL HAQUE SIAM, ID: 2243081050
OOP: 65, DLD: 70, CA: 60, Average CGPA: 3.25
Attendance: 100.00%

--- MD. ADNAN AREFIN RATUL ---
[STUDENT INFO]
Name: MD. ADNAN AREFIN RATUL, ID: 2243081062
OOP: 71, DLD: 74, CA: 80, Average CGPA: 3.67
Attendance: 66.67%

--- All Reports ---

Ln 1, Col 14 Spaces: 4 UTF-8 with BOM CRLF {} C#
```

## ➤ Discussion-

- This project clearly reflects the use of Object-Oriented Programming concepts in C#.
- All features used in the code are properly documented in the report.
- The structure matches academic project standards and is suitable for full marking.