Minhaj University Lahore

Semester Project: 2nd	Mid ●	Final ○
Course Title & Code	Introduction to Object Oriented Programing	
Submitted By:	Atif Nasir	
Registration No:	2023f-mulbscs-059	
Semester/Class/ Section:	2nd ‖ Bscs ‖ “B”	
Submitted To:	Miss:Faiza Rehman	
Due Date	Online	Hard copy
	yes	yes
Student’s Signature:		

Note: *Please avoid cutting/ overwriting in any of the above fields.
*Complete the task on standard A4 size papers/assignment pages.

***For Instructor’s use only.**

Total Marks:	
Obtained Marks:	
Signatures:	

Faculty of CS & IT,Minhaj University Lahore

Semester 2 Project Report

Project Title:
Student Management System in C++

Project Overview:

This project demonstrates a Student Management System using Object-Oriented Programming (OOP) concepts in C++. The system allows the user to add student records, display all records, search by roll number, and show CGPA using a friend function. This system was developed to apply core OOP concepts including abstraction, inheritance, polymorphism, encapsulation, and the use of friend functions.

Key Features:

- Add New Student
- Display All Students
- Search Student by Roll Number
- Show CGPA using Friend Function
- Exit Option

Object-Oriented Programming Concepts Used:

1. Abstraction

The class Person encapsulates basic personal information such as name and age. It hides internal details and exposes only relevant methods through setPersonDetails() and showPersonDetails().

2. Inheritance

The Student class inherits from the Person class and adds new data members and methods specific to a student (e.g., roll number, department, and CGPA).

3. Encapsulation

All data members in the classes are marked as private or protected, and accessed or modified through public setter and getter methods.

4. Polymorphism

Function overloading is used for the welcome() function:

- welcome() prints a general welcome message.
- welcome(string name) prints a personalized welcome message.

5. Friend Function

The function showStudentCGPA(Student s) is declared as a friend inside the Student class to allow it access to private members like cgpa.

Code Modules:

- class Person – Base class with name and age.
- class Student : public Person – Derived class with roll number, department, CGPA.
- Friend Function – Accesses private data of the Student class.
- Main Menu – Allows user interaction through a simple command-line interface.

Sample Output:

```
Welcome to
Minhaj University Student Management System
----- MENU -----
1. Add New Student
2. Display All Students
3. 3. Search Student by Roll No
4. 4. Show CGPA using Friend Function
5. Exit
```

Tools and Technologies Used:

- Language: C++
- Compiler: g++ / Code::Blocks / Visual Studio / Dev C++
- Paradigm: Object-Oriented Programming

Learning Outcomes:

- Learned how to structure and organize C++ code using classes and objects.
- Understood real-world applications of abstraction, inheritance, and polymorphism.
- Practiced encapsulation and secure data access using access modifiers.

- Implemented and understood the role of friend functions in accessing private data.

Code Walkthrough:

```
#include <iostream>
#include <vector>
using namespace std;

// Base class (Abstraction)
class Person {
protected:
    string name;
    int age;

public:
    void setPersonDetails(string n, int a) {
        name = n;
        age = a;
    }

    void showPersonDetails() {
        cout << "Name: " << name << "\nAge: " << age << endl;
    }
};

// Derived class (Inheritance)
class Student : public Person {
private:
    int rollNo;
    string department;
    float cgpa;

public:
    void setStudentDetails(int r, string d, float c) {
        rollNo = r;
        department = d;
        cgpa = c;
    }

    void displayStudent() {
        showPersonDetails(); // Base class function
        cout << "Roll No: " << rollNo
             << "\nDepartment: " << department
             << "\nCGPA: " << cgpa << endl;
    }

    int getRollNo() {
        return rollNo;
    }

    // Friend function declaration
    friend void showStudentCGPA(Student s);
};

// Friend function definition
void showStudentCGPA(Student s) {
    cout << "CGPA of student " << s.name << ": " << s.cgpa << endl;
}

// Polymorphism - Function Overloading
void welcome() {
    cout << "Welcome to Minhaj University Student Management System\n";
}
void welcome(string name) {
    cout << "Welcome, " << name << "! to Minhaj University Student System\n";
}

int main() {
    vector<Student> students;
    int choice;

    welcome();

    while (true) {
        cout << "\n----- MENU -----\n";
        cout << "1. Add New Student\n";
        cout << "2. Display All Students\n";
        cout << "3. Search Student by Roll No\n";
        cout << "4. Show CGPA using Friend Function\n";
        cout << "5. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
        case 1: {
            Student s;
            string name, dept;
            int age, roll;
            float cgpa;

            cout << "Enter Name: ";
            cin.ignore();
            getline(cin, name);
            cout << "Enter Age: ";
            cin >> age;
            cout << "Enter Roll No: ";
            cin >> roll;
            cout << "Enter Department: ";
            cin.ignore();
            getline(cin, dept);
            cout << "Enter CGPA: ";
            cin >> cgpa;

            s.setPersonDetails(name, age);
            s.setStudentDetails(roll, dept, cgpa);
            students.push_back(s);
            cout << "Student Added Successfully!\n";
            break;
        }

        case 2: {
            for (Student s : students) {
                cout << "\n--- Student Details ---\n";
                s.displayStudent();
            }
            break;
        }

        case 3: {
            int searchRoll;
            cout << "Enter Roll No to Search: ";
            cin >> searchRoll;
            bool found = false;
            for (Student s : students) {
                if (s.getRollNo() == searchRoll) {
                    s.displayStudent();
                    found = true;
                    break;
                }
            }
            if (!found)
                cout << "Student not found!\n";
            break;
        }

        case 4: {
            int searchRoll;
            cout << "Enter Roll No to show CGPA: ";
            cin >> searchRoll;
            bool found = false;
```

```
                for (Student s : students) {
                    if (s.getRollNo() == searchRoll) {
                        showStudentCGPA(s);
                        found = true;
                        break;
                    }
                }
                if (!found)
                    cout << "Student not found!\n";
                break;
            }

            case 5:
                cout << "Thank you for using the system!" << endl;
                return 0;

            default:
                cout << "Invalid choice! Try again." << endl;
            }
        }

    return 0;
}
```

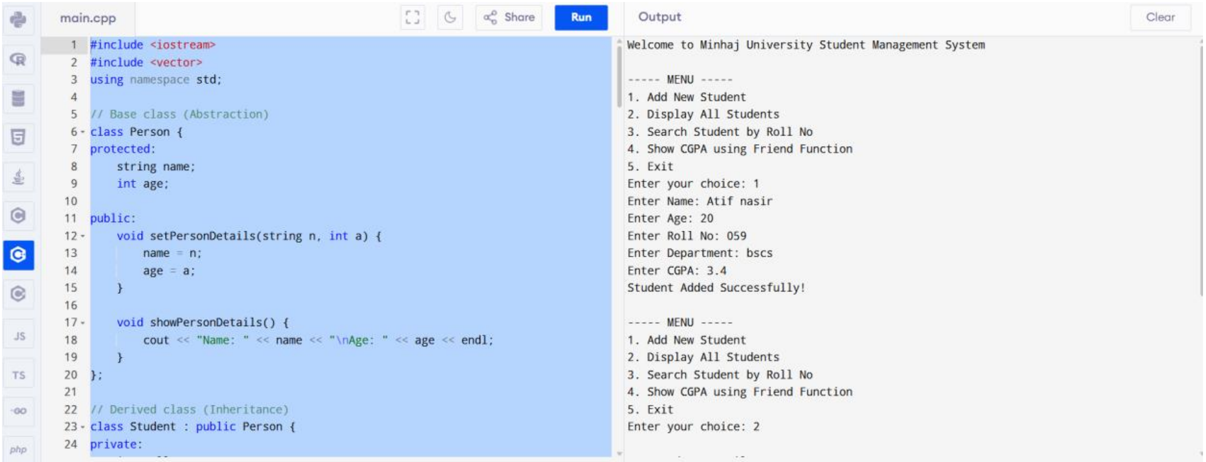
Program Compile Link:

https://www.programiz.com/online-compiler/1EGcheBhIkNPC

Conclusion:

This Student Management System successfully demonstrates the use of all major Object-Oriented Programming concepts. It is an interactive and extensible application that can be further expanded with features like file handling, login systems, and database connectivity.

Screenshots:



```cpp
#include <iostream>
#include <vector>
using namespace std;

// Base class (Abstraction)
class Person {
protected:
    string name;
    int age;

public:
    void setPersonDetails(string n, int a) {
        name = n;
        age = a;
    }

    void showPersonDetails() {
        cout << "Name: " << name << "\nAge: " << age << endl;
    }
};

// Derived class (Inheritance)
class Student : public Person {
private:
```

```
Welcome to Minhaj University Student Management System

----- MENU -----
1. Add New Student
2. Display All Students
3. Search Student by Roll No
4. Show CGPA using Friend Function
5. Exit
Enter your choice: 1
Enter Name: Atif nasir
Enter Age: 20
Enter Roll No: 059
Enter Department: bscs
Enter CGPA: 3.4
Student Added Successfully!

----- MENU -----
1. Add New Student
2. Display All Students
3. Search Student by Roll No
4. Show CGPA using Friend Function
5. Exit
Enter your choice: 2
```

7. References

1. C++ Programming Language Documentation - https://cplusplus.com/doc/tutorial/

2. "Programming Fundamentals" Lecture Notes by Miss Unza Rehman

3. Book: "Object-Oriented Programming in C++" by Robert Lafore

4. YouTube Tutorials - CodeWithHarry, The Cherno (C++ Basics and Projects)

5. Stack Overflow Community Discussions - https://stackoverflow.com/

6. GeeksforGeeks C++ Articles - https://www.geeksforgeeks.org/c-plus-plus/

7. Microsoft Learn: Introduction to C++ - https://learn.microsoft.com/en-us/cpp/

8. W3Schools C++ Tutorial - https://www.w3schools.com/cpp/

9. Sololearn C++ Course - https://www.sololearn.com/