BAHRIA UNIVERSITY

OBJECT ORIENTED PROGRAMMING

NAME: KASHAN MAQSOOD

ENROLLMENT: 01-136232-022

CLASS: BS AI 2-A

SEMESTER PROJECT

FACULTY NAME : SIR ADIL KHAN

DEPARTMENT OF COMPUTER SCIENCE

BAHRIA UNIVERSITY ISLAMABAD CAMPUS

Student Management System in C++

1. Introduction:

The **Student Management System** is a console-based application developed in **C++** to streamline the management of student records. It provides functionalities such as adding, searching, updating, and deleting student details, along with data persistence through file handling. This system enhances efficiency, reduces manual effort, and ensures accurate record-keeping. The project applies **object-oriented programming (OOP)** principles, input validation, and error handling to ensure reliability and scalability.

2. Objectives

The main objectives of this project are:

- To develop a **simple yet effective** system for managing student records.
- To implement **object-oriented programming** concepts in C++.
- To provide **efficient data handling** with the use of vectors and file management.
- To ensure **data persistence** by saving and retrieving student records from a file.
- To enhance user experience through a **simple and intuitive console interface**.
- To apply **error handling and input validation** for improved accuracy and robustness.

3. Methodology

The development process of the **Student Management System** follows these steps:

1. **Requirement Analysis** – Understanding the key features required for managing student records.

2. **System Design** – Structuring the program using **OOP principles** (classes, objects, and encapsulation).
3. **Implementation** – Writing C++ code using standard libraries (iostream, fstream, vector, and string).
4. **Testing & Debugging** – Ensuring correct functionality, validating inputs, and handling errors.
5. **Data Persistence** – Implementing **file handling** for saving and retrieving student records.
6. **Finalization** – Optimizing performance and documenting the system for future scalability.

4. Features

The **Student Management System** includes the following features:

- **Add New Student** – Allows users to store student details such as name, enrollment number, and grades.
- **Search Student** – Finds a student by **enrollment number** for quick access.
- **Update Student Information** – Modifies existing student details.
- **Delete Student Record** – Removes a student's data from the system.
- **Display All Students** – Lists all stored student records.
- **Calculate Average Grade** – Computes the overall class performance.
- **File Handling for Data Persistence** – Saves student records to a file and loads them when needed.

5. Outcomes

By successfully developing and implementing this system, the following outcomes were achieved:

- **Efficient Student Data Management** – Streamlined record-keeping, reducing manual errors.

- **Improved Programming Skills** – Strengthened knowledge of **C++**, **OOP principles**, and **file handling**.
- **Reliable Data Handling** – Ensured data integrity through validation and error handling.
- **Practical Application of Concepts** – Applied theoretical knowledge to develop a functional and useful system.

6. Conclusion

The **Student Management System** is a well-structured application that effectively manages student records using **C++**. The project demonstrates the practical use of **OOP, file handling, and error management**. It serves as a fundamental yet scalable solution that can be extended in the future.

7. Future Enhancements

To further improve the system, the following enhancements can be implemented:

- **Graphical User Interface (GUI)** – Developing a user-friendly GUI for better usability.
- **Database Integration** – Using **MySQL** or **SQLite** for scalable and efficient data storage.
- **Advanced Search & Filters** – Implementing **multiple search filters** for better data retrieval.
- **User Authentication** – Adding **login/logout functionality** for security.
- **Reports & Analytics** – Generating **detailed reports** for student performance analysis.