Addis Ababa Science and Technology University

Fundamentals of Programming II (C++)

Project Work Guideline and Description

---

1. Introduction

The purpose of this project is to enable students to apply the programming concepts learned in **Fundamentals of Programming II** through the design and development of a real-world C++ application. Students are expected to demonstrate problem-solving skills, teamwork, software development best practices, and technical documentation.

The project shall be completed in groups using collaborative development tools and shall conclude in a working software application, documentation, and project presentation.

2. Learning Objectives

Upon successful completion of the project, students should be able to:

- Analyze real-world problems and understand system requirements.
- Design algorithms and program structures.
- Develop a complete C++ implementation.
- Implement appropriate programming techniques.
- Use GitHub for collaborative application development.
- Produce professional technical documentation.
- Present and defend their work effectively.

3. General Project Requirements

The project implementation must demonstrate the following programming concepts:

- Structures and/or Classes
- Functions
- Arrays
- Pointers
- Function/class Templates
- File Handling
- Menu-Driven Programming
- Object-Oriented Programming Principles (if applicable)

4. Functional Requirements

The project must provide the following minimum functionalities:

a. **User-friendly Menu System** that allows users to navigate and perform operations easily

b. The system shall support the following **Data Management Operations**:

- Add new records
- Display records in tabular format
- Update existing records
- Delete individual records
- Delete all records (with confirmation)
- Search records using relevant keys
- Sort records in ascending and descending order

c. The application must allow user **save data to files** and **load data from files**

d. The system must support the following **Validation and Error Handling**

- Validate user inputs
- Handle invalid menu selections
- Prevent duplicate records where applicable
- Handle file-related errors gracefully

e. Students are expected to follow **programming best practices** including:

i. **Coding Standards**	ii. **Documentation Standards**
• Meaningful variable names • Meaningful function names • Consistent indentation • Proper code formatting • Modular design • Appropriate comments • Avoidance of code duplication	• Well-organized • Properly formatted • Consistent in font style and size • Free from language errors

5. Project Deliverables

a. Deliverable 1: Source Code

- Complete C++ source code
- Well-structured folder organization
- Proper commenting

b. Deliverable 2: Project Documentation

- A comprehensive project report in PDF format.

c. Deliverable 3: Presentation Slides

- Presentation summarizing problem definition, design, implementation, challenges and lessons learned

d. Deliverable 4: Demonstration

- Live demonstration of the application during project defense.

6. Documentation Structure

The project report should contain the following sections:

1. Cover Page
2. Executive Summary
3. Problem Description
4. Requirement Analysis including IPO (Input-Process-Output) Analysis
5. Algorithm Design: Flowcharts, Pseudocode and
6. Description of Program Implementation
7. Challenges and Solutions
8. Conclusion and Recommendations
9. References

7. Evaluation Rubric

No	Evaluation Component	Weight (%)
1	**Problem Analysis & Design**	15
2	**Program Implementation**	30
3	**Required Programming Concepts**	20
4	**Documentation Quality**	10
5	**GitHub Contribution**	10
6	**Testing & Validation**	5
7	**Presentation & Demonstration**	10
	Total	**100**

8. Academic Integrity

Students are expected to maintain academic honesty.

The following are prohibited:

- Copying another group's project.
- Submitting downloaded projects as original work.
- Plagiarizing documentation.
- Misrepresenting contribution levels.

Violations may result in grade penalties or project rejection.

9. Problem Description

9.1. Project Title:

Student Record Management System

9.2. Background

Colleges manage large volumes of student academic information, including personal details, course registrations, grades, and academic status. Manual management of these records is inefficient and increases the risk of errors.

A computerized Student Record Management System can streamline academic record management and improve decision-making through automated grade computation and reporting.

9.3. Problem Statement

The department currently stores student academic records manually, making it difficult to:

- Track student performance.
- Calculate cumulative grades.
- Determine academic standing.
- Generate class rankings.
- Produce statistical reports.

The department requires an automated system capable of managing student records efficiently and accurately.

9.4. Project Description

You are tasked with C++ application development for Student Record Management System that manages student information, course registration, grades, and academic performance.

The system should support:

- Student registration.
- Course registration.
- Grade management.
- GPA/CGPA calculation.
- Student ranking.
- Academic status determination.
- Statistical reporting.

All records should be stored in files for long-term access and retrieval.

9.5 Functional Requirements

1	Student Management	• Register students. • Edit student records. • Delete student records. • Search students.
2	Course Management	• Register courses. • Update course information. • Search courses. • Check prerequisite requirements
3	Grade Management	• Record grades. • Update grades. • Compute semester GPA. • Compute cumulative CGPA
4	Academic Evaluation	• Determine student status: Pass, Warning, Dismissal • Generate student ranking. • Identify top-performing students
5	Reporting	• Number of students. • Pass rate. • Warning rate. • Department statistics. • CGPA distribution

9.6. Expected Outputs

The system should generate:

- Student transcript.
- Semester report.
- CGPA report.
- Academic status report.
- Ranking report.
- Statistical summaries.