

PROPOSAL
UNIVERSITY MANAGEMENT SYSTEM



Name: M Awais

Enrollment Number: 09-136242-029

Project Proposal: Student Management System (SMS)

Project Title: Student & Academic Record Management System using Dynamic Memory Allocation **Language:** C++ **Domain:** Console Application / Data Structures

1. Introduction

The **Student & Academic Record Management System** is a C++ console-based application designed to streamline the administration of educational records. Unlike traditional static systems, this project utilizes **Object-Oriented Programming (OOP)** and **Dynamic Memory Allocation** to manage data efficiently during runtime. The system serves as a central interface for managing details regarding students, teachers, courses, and academic grades.

2. Problem Statement

In many basic computing systems, memory is allocated statically (using fixed-size arrays). This leads to two major issues:

1. **Memory Wastage:** If the reserved space is not fully used.
2. **Memory Shortage:** If the data exceeds the pre-defined limit. Furthermore, manual record-keeping on paper is prone to errors, data loss, and inefficiency in retrieval. There is a need for a computerized system that manages memory flexible based on the exact number of records required.

3. Objectives

The primary objectives of this project are:

- To develop a system that allows users to **Create, Read, Update, and Delete (CRUD)** academic records.
- To implement **Dynamic Memory Allocation** (using pointers and the heap) to ensure memory is only used when necessary.
- To demonstrate the relationship between different academic entities (Students, Courses, Teachers) using OOP classes.

- To ensure efficient memory management by implementing proper deallocation techniques to prevent memory leaks.

4. Scope of the Project

The project is designed to cover the following areas:

- **Student Module:** Management of personal details and enrollment dates.
- **Teacher Module:** Handling of faculty data, departments, positions, and salaries.
- **Academic Module:** Management of course codes, credits, and instructors.
- **Grading Module:** Linking students to courses and assigning grades.
- **Memory Management:** The system will operate in the computer's RAM using heap memory.

Note: This version runs in the console and stores data temporarily during the execution of the program.

5. Methodology & Technical Approach

The project is built using the **C++ Programming Language**.

- **Paradigm:** Object-Oriented Programming (OOP).
- **Memory Model:** The system exclusively uses **Pointers (*)** and **Dynamic Allocation** (new keyword) for all data members (Strings, Integers) and objects.
- **Data Structure:** Arrays of Pointers are used to manage lists of students and courses, allowing for index-based manipulation.
- **Cleanup:** The system implements Destructors (~Class) and the delete operator to manually free memory, simulating low-level system resource management.

6. Functional Requirements (Features)

1. **Data Entry:** Users can dynamically specify the number of students to register at runtime.

2. **Bio-Data Management:** Detailed input/output for Student and Teacher profiles.
3. **Course & Grading:** Ability to link courses to students and assign performance grades.
4. **Update Record:** Users can modify specific records (Student, Course, or Grade) without affecting others.
5. **Delete Record:** Users can remove specific entries. The system immediately frees the memory associated with that specific entry to optimize resources.

7. Conclusion

This Student Management System provides a robust framework for understanding how academic data can be structured programmatically. By utilizing dynamic memory allocation, the project offers a technical advantage over static array-based systems, providing a foundation for scalable memory management in software development.