

THE UNIVERSITY OF AZAD JAMMU AND KASHMIR, MUZAFFARABAD



COURSE TITLE	OOP
COURSE CODE	CS-1204
PROJECT TITLE	Smart Task Manager
STUDENT NAME	Mahdi Ali, Tayyab Hanif, Zarnab
STUDENT ROLL NUMBERS	2024-SE-16, 2024-SE-11, 2024- SE-14
INSTRUCTOR NAME	Engr. Awais Rathore
SUBMISSION DATE	September 26, 2025

1. Introduction

The Smart Task Management System is a console-based C++ application designed to help users efficiently manage their daily tasks. The system provides features such as adding, updating, deleting, and marking tasks as completed, with validations to ensure proper data input. The program also uses file handling to persist data across multiple runs, ensuring that tasks are saved and restored automatically.

The project is built using Object-Oriented Programming (OOP) principles and demonstrates practical use of constructors, destructors, copy constructors, static members, friend classes, exception handling, as well as concepts like composition and aggregation.

2. Objectives

- To provide a simple and effective tool for managing tasks.
- To demonstrate the application of OOP concepts in C++.
- To include file handling for persistent storage of data.
- To make the console interface user-friendly, neat, and visually appealing.
- To ensure robust input validation, avoiding invalid or inconsistent data.

3. Features of the System

- Add Task – Add tasks with title, description, priority, and time. Validations prevent duplicate timings for pending tasks.
- View Tasks – Display tasks in detailed or summary table view with colors. Pending and completed tasks highlighted differently.
- Update Task – Update details of an existing pending task. Completed tasks cannot be updated.
- Delete Task(s) – Delete one or multiple tasks at once by entering task IDs. Task IDs are rearranged automatically.
- Mark Task(s) as Completed – Mark one or more tasks as completed. Completed tasks are skipped and reported back.
- File Handling – Save tasks to a file automatically and reload on restart.
- Validation – Input validation for integers, hours/minutes, IDs. Prevention of duplicate pending task timings.
- Console Formatting & Coloring – Use of Windows.h for text coloring and neat menu formatting.

4. Technologies Used

Language: C++

IDE: Dev-C++

Libraries Used:

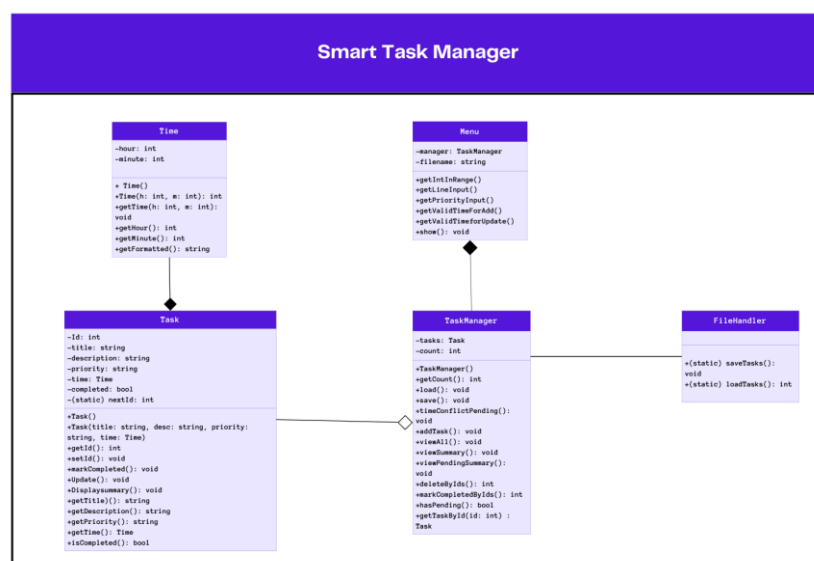
- <iostream> for input/output
- <fstream> for file handling
- <stdexcept> for exception handling
- <string> for strings
- <iomanip> for formatted output
- <windows.h> for console coloring

5. Object-Oriented Concepts Applied

- Classes & Objects – Divided into multiple classes (Task, Time, TaskManager, FileHandler, Menu).
- Composition – Task has-a Time.
- Aggregation – TaskManager has-many Task objects.
- Composition – Menu owns TaskManager.
- Friendship – FileHandler can access private members of Task.
- Static Members – Used in Task for auto-incrementing task IDs.
- Copy Constructor – Implemented for copying Task objects.
- Exception Handling – Used for invalid time or duplicate tasks.
- Encapsulation – Attributes are private with public getters/setters.

6. Class Diagram

- Menu ◆— TaskManager (composition)
- TaskManager ◇— Task (aggregation)
- Task ◆— Time (composition)
- FileHandler → Task (Association)



7. Conclusion

The Smart Task Management System demonstrates how C++ can be used to build a structured and user-friendly console application. The project successfully integrates multiple OOP concepts, file handling, and console enhancements to provide a practical solution for task management.

This project was not only an exercise in coding but also in software design and documentation. The systematic breakdown into classes, use of diagrams, and application of theory into practice reflects a complete software