Academic management system:
By Malaika Nasir

Main functionalities:

Following are the main functionalities which I have used in my code:

Inheritance: Inheritance is a concept in object-oriented programming that allows you to create a new class based on an existing class. It enables you to reuse the code of the base class and add or modify the behavior as needed. Inheritance also enables polymorphism, which allows you to treat objects of different classes as if they were of the same class. Encapsulation is another advantage of inheritance, as it allows you to hide the implementation details of a class. Finally, inheritance can make code maintenance easier by allowing changes to be made to the base class, which are automatically inherited by all derived classes.

Aggregation: Aggregation is another concept in object-oriented programming that allows you to build complex objects from simpler objects. It enables you to create a class that contains one or more instances of another class. Aggregation is a "has-a" relationship, meaning that the containing class "has-a" reference to the contained class. This allows you to reuse the code of the contained class and create more complex and flexible software systems.

Composition: In functional programming, composition is a similar concept to aggregation in object-oriented programming. Composition allows you to build complex functions from simpler functions by chaining them together. This is done by passing the output of one function as the input to another function. Composition enables you to create more flexible and reusable functions, as well as simplify the code by breaking it down into smaller, more manageable parts. Additionally, composition allows you to abstract away the details of how a function is implemented, making it easier to reason about the behavior of the function as a whole.

Libraries:

Following are the libraries which I have used in the code:

#include <fstream> : The "#include <fstream>" is a preprocessor directive in C++ that includes the "fstream" library, which provides functions for reading and writing to files. This allows you to use file I/O operations in your C++ programs.

#include <string>: The "#include <string>" is a preprocessor directive in C++ that includes the "string" library, which provides functions and classes for working with strings of characters. This allows you to create and manipulate strings, as well as perform various string operations, such as concatenation and substring extraction, in your C++ programs.

#include <sstream>: The "#include <sstream>" is a preprocessor directive in C++

that includes the "sstream" library, which provides functions and classes for working with string streams. This allows you to read from and write to strings as if they were input/output streams. It also provides facilities for converting between strings and other data types, such as integers and floating-point numbers.

#include <conio.h>: The "#include <conio.h>" is a preprocessor directive in C and C++ that includes the "conio" library, which provides functions for performing console input and output operations. This library is typically used in Windows environments to access console-specific functions, such as reading keystrokes and setting console colors. However, it should be noted that the use of the "conio" library is not portable and may not work on all platforms.

#include <ctime>: The "#include <ctime>" is a preprocessor directive in C++ that includes the "ctime" library, which provides functions for working with date and time values. This library provides facilities for obtaining the current time, converting between different time formats, and performing various time-related calculations. It also includes functions for working with the system clock and measuring time intervals.

MY IMPLEMENTATION:

Class user has functions which helps with the login system i.e checking password and giving a default password . In the function isvalidpass it is checked weather the password has 6 length, has a lowercase and uppercase character and digit and special character. It can also modify the password.

Class teacher check if the id entered is correct or not. It also tells the teachers course. This class inherits class user so that it can modify or check the password.

Class student checks if the id entered is correct is correct or not. Then it offers the courses only I which the student is registered by loops. It also show that there is invalid entry if your student id is not in the form 2xI-xxxx where x can be any digit. This class inherits class user so that it can modify or check the password.

Class course checks if the student is registered in the course or not. It inherits the class student to take student information to check for its course.

Class time first take the time input from teacher and then it checks if the time is between the current time or not by if statement and a lot of and checks. If the time is between the current time it returns true otherwise the student cannot attempt the quiz.

Class date first take the date input from teacher and then it checks if the date is within the current date or not by if statement and a lot of and checks. If the date is within the current date it returns true otherwise the student cannot attempt the quiz.

Class Quiz inherits student teacher and course class, it first checks which student is registered in which course then it helps with choosing the file for the quiz with switch and cases. The default shows that no registered course has been found.

Main has a work which allows the user to input its id and password and while password is being entered there is a loop showing asterisk which runs until entered is pressed then this loop ends and the characters are saved in the variable. It then checks if the password is correct or not.