**OOP PROJECT :**

PRESENTED BY :

1. MUHAMMAD TALHA-46287
2. MARWA JABBAR-47129

SUBMITTED TO : MA’AM MADEEHA

SUBJECT : OOP

SEC/SEM : 2-A

**TOPIC:**

Gym Management System.

**Table of Contents :**

* [What is Gym Management System?](https://learnprogramo.com/gym-management-system-project-in-c-download-with-source-code/#What_is_Gym_Management_System)
* [Features of Gym Management System Project in C++](https://learnprogramo.com/gym-management-system-project-in-c-download-with-source-code/#Features_of_Gym_Management_System_Project_in_C).
* [Modules of Gym Management System Project in C++](https://learnprogramo.com/gym-management-system-project-in-c-download-with-source-code/#Modules_of_Gym_Management_System_Project_in_C).
* [Working of Project](https://learnprogramo.com/gym-management-system-project-in-c-download-with-source-code/#Working_of_Project)
  + [Main Screen:](https://learnprogramo.com/gym-management-system-project-in-c-download-with-source-code/#Main_Screen)
  + [1. User](https://learnprogramo.com/gym-management-system-project-in-c-download-with-source-code/#1_User).
  + [2. Admin](https://learnprogramo.com/gym-management-system-project-in-c-download-with-source-code/#2_Admin).
* [Summary:](https://learnprogramo.com/gym-management-system-project-in-c-download-with-source-code/#Summary)

**What is Gym Management System?**

As the name suggests the Gym Management System is a software that handles the entire data of the Gym. It makes the work of a Fitness Manager very easy instead of writing data in a notebook. In past, the gym managers were using notebooks to write the data along with customers’ names who purchased the gym memberships. So it was very difficult to keep track of each and every record.

If a Gym Manager wants to search for a particular gym record then that task was very time-consuming. So to make this task easy the programming languages were developed and C++ language is one of them.

**Features of Gym Management System Project in C++**

* We have created separate logins for the users and the gym managers i.e admin, in which the gym manager is password protected.
* In this project, the gym manager can add, update, delete and create gym memberships and can assign them to a particular user.
* The user can quit the gym anytime, whenever he or she wants.
* The software gives facility for the users to edit their profiles.
* Only the admin can delete and edit the membership.
* During the purchase of membership, the trainer is allotted automatically.
* This project uses the concept of file handling.

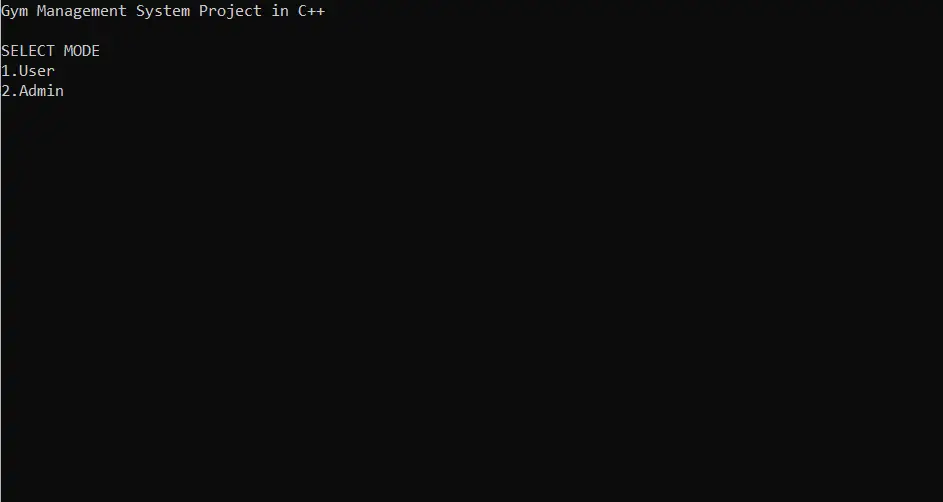
**Modules of Gym Management System Project in C++**

* Join Gym.
* Quit Gym.
* Edit User Profile.
* Create Membership
* Display All Records
* Search Particular Record
* Edit Membership Details
* Delete Membership

**Working of Project:**

* **Main Screen:**

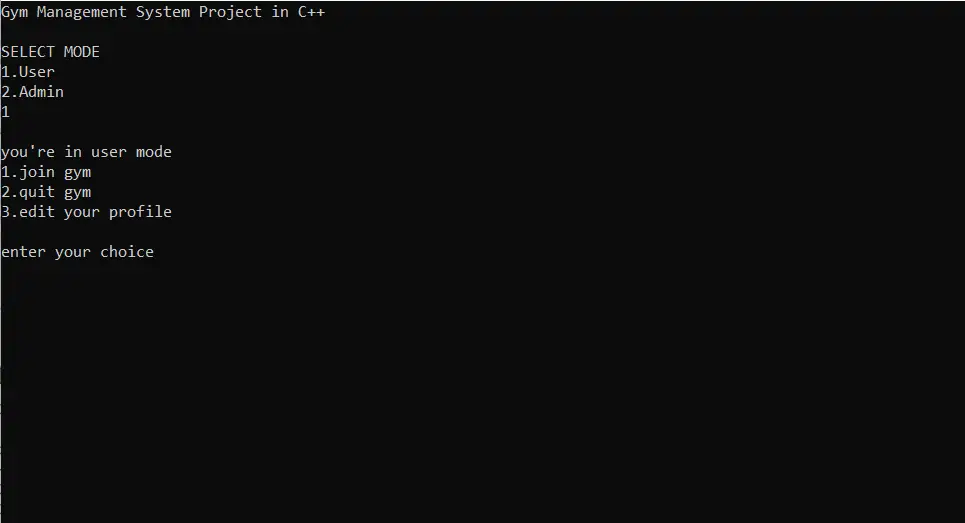
When you run the project from any compiler or directly click on the executable .exe file you’ll see the following screen shown in the picture.



We’ve displayed the menu of User and the Admin. If you’re a user then your choice will be 1 and if you are a gym manager or admin then your choice will be 2. Now we’ll discuss each and every menu in detail.

### 1. User:

The user will not require the additional sign-in, he or she will be able to access the software directly. When the user the choice as 1 then the following screen wilL appear:



**1. Join Gym:**

With the help of this menu option, the new users who want to join the gym can register themself. After choosing the option 1 the software will ask the user to enter the following details:

* The new member number user wanted.
* Name of the new member.
* Phone number.
* Gold or Silver membership.
* Preferred time slot.

After entering all the above information the trainer will get allotted to the user and also the registration will be done successfully.

**2. Quit Gym:**

If the user’s mind changes or the user wanted to change the gym then he or she can quit the gym anytime by choosing the menu option 2. After choosing menu option 2 the software will ask you to enter the member number. And in a fraction of a few seconds, the member will be deleted from the software.

**3. Edit your Profile:**

This software comes with a unique facility, where the users are able to edit their profiles. If the user wants to edit their profile then they should choose the menu option 3. After choosing the menu option 3 the software will ask you to enter your membership number by which the software will fetch all the details from the file. Then the user has to enter all the details that they want to edit. After entering all the information the profile will be successfully updated.

**2. Admin:**

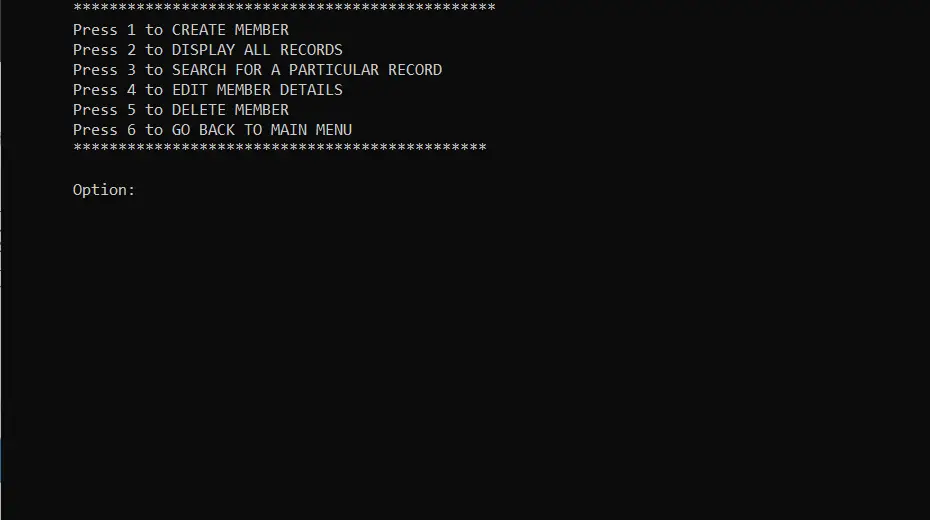
To access the features of the admin menu, He or she will require to sign in using the username “admin” and the password which is “**learnprogramo**“. We’ve also given the facility to change the password in the Admin menu. Only Admin has rights to change the password. When the user presses the choice as

2. Then the software will ask you to enter the correct password as shown in the following image:



If the password is incorrect the application will show the error of the wrong password. And if the password is correct then the admin menu will be visible to the user where he or she can do the operations displayed in the menu.

The following menu will be visible to the Gym Manager:



**1. Create Member:**

If the gym manager wants to add a membership for the new joiner then he should select the menu option 1. After selecting 1st option the software will ask admin to enter information such as Name of the member, Enter contact number, Gold or Silver Membership and Preferred time slot. After entering all the information the trainer will be allocated automatically and the record will be added successfully.

**2. Display All Records:**

If the gym manager wants to display all the records or the memberships then after selecting the 2nd option all the records will be displayed. The information such as Member code, Name, Category, Fee, Contact and the Timings will be displayed.

**3. Search Particular Record:**

This software comes with the amazing feature such as searching a record in the fraction of seconds. If the gym manager wants to search for a specific record then by choosing the 3rd option he’ll be able to search the record…

**4. Edit Member Details:**

The admin has all the rights to edit the details of the member user. If the gym manager found some mistake in the member information then by choosing this option he or she will be able to edit all the record of existing user.

**5. Delete Member:**

If the membership of any specific user ends then there should be a facility to delete the membership. So we’ve added functionality of deleting the member record by choosing 5th option. After choosing the 5th option the software will ask admin to enter the member number so that the software can search the specific record to delete. After entering the correct member number the record will be deleted successfully.

**Key Points/Functions Used:**

The provided code is a program for a gym management system. It allows users to join the gym, manage member records, and perform administrative tasks. Let's break down the code and understand its structure and functionality.

The code begins by including necessary header files and namespaces for input/output operations and standard library functions. The headers included are `fstream`, `cstring`, `iomanip`, `iostream`, and `cstdlib`.

Next, a class named `login` is defined. It has a private member variable `admin` and two member functions `adlogin()` and `getadpassword()`. The `adlogin()` function prompts the user to enter an admin code and checks if it matches the expected value. If the code is correct, it calls the `getadpassword()` function to verify the admin password. If both the code and password are correct, it displays a success message; otherwise, it displays appropriate error messages and allows the user to retry.

The `getadpassword()` function is responsible for obtaining the admin password from the user. It checks if the entered password matches the expected value. If the password is correct, it displays a welcome message; otherwise, it displays an error message and allows the user to retry.

Next, there are two functions `allotment\_gold()` and `allotment\_silver()`, which handle the allotment of trainers for gold and silver classes based on the availability of trainers.

The `trainee\_allotment()` function takes a character pointer as a parameter and checks the availability of trainers for the specified class type (gold or silver). If trainers are available, it calls the respective allotment function (`allotment\_gold()` or `allotment\_silver()`). If no trainers are available, it displays an appropriate message.

After that, a class named `member` is defined, which inherits from the `login` class. It has several member variables such as `member\_number`, `mem\_name`, `classs`, `timings`, `fee`, and `contact`. It also has member functions like `time\_slots()`, `create\_mem()`, `show\_mem()`, and various getter functions.

The `time\_slots()` function prompts the user to select preferred timings for the gym. Based on the user's input, it assigns a value to the `timings` member variable.

The `create\_mem()` function collects information from the user such as member number, name, contact number, class type (gold or silver), and preferred timings. It calculates the monthly fee based on the class type and displays it. It then calls the `trainee\_allotment()` function to check the availability of trainers for the selected class.

The `show\_mem()` function displays the member's information such as member number, name, class type, fee, contact number, and preferred timings.

Next, the code defines a global `fstream` object named `fp` and an object of the `member` class named `m1`.

The `save\_member()` function opens the file "gymdata.txt" in append mode and calls the `create\_mem()` function to get member information. It then writes the `m1` object to the file and displays a success message.

The `show\_all()` function displays all the records stored in the "gymdata.txt" file by reading the file and calling the `show\_mem()` function for each record.

The `display\_record()` function takes a member number as input and searches for the corresponding record in the file. If found, it displays the record using the `show\_mem()` function.

The `edit\_member()` function allows the user to edit the details of a member by entering the member number. It searches for the record in the file, displays the existing details, prompts for new details using the `create\_mem()` function.

---------------------------------------**THANK YOU**-------------------------------------------------