INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, LUCKNOW

SCHOOL MANAGEMENT SYSTEM

PROJECT REPORT

LIT2020008	AKSHAY UPADHYAY
LIT2020023	ADITYA AGARWAL
LIT2020034	ANKIT KUMAR
LIT2020036	PRABHAV AGARWAL

Teacher: Saurabh Shrivastav Sir

IT

CONTENTS

INTRODUCTION / MAIN FEATURES OF OUR PROJECT

The topic of our project is “School Management System”. Through this project we have tried to replicate major functions that take place in school. We have tried to keep our project as close to the real world as possible.

Basically we have made our project for 7 different users namely

User 1 : ADMIN – the person who will manage the entire SCHOOL MANAGING SYSTEM

User 2 : TEACHER – this will provide functionality to all the teachers added to the system

User 3: STUDENT – this will provide functionality to all the student added to the system

User 4: STAFF – this will provide functionality to all the staff added to the system

User 5: CLASS_MONITOR – this will provide functionality to all the class monitors added to the system

User 6: CLASS TEACHER – this will provide functionality to all the class teachers added to the system

User 7: LEADER – the head boy/girl (basically the leader of all the class monitors)

Apart from providing diverse functionality to the 7 uses, our system also maintains two important entities without which no school can function

A. Bank management system → managing the school amount details
B. Library management system → managing the school library

In order to implement our project we have used Object Oriented Programming which uses the concepts of Object Oriented Methodology.

Language used → C++

Compiler used → g++

Editor used → VScode

WHY OOP, not POP

Procedure-Oriented Programming

Conventional programming, using high-level languages such as COBOL, FORTRAN and C, is commonly known as procedure oriented programming (POP).

Problems with POP

- In the procedure-oriented approach, the problem is viewed as a sequence of things to be done such as reading, calculating and printing.
- In a large program it is very difficult to identify what data is used by which function.
- Another serious drawback with the procedural approach is that it does not model real world problems very well. This is because functions are action oriented and do not really corresponds to the elements of the problem

Object–Oriented Programming

The major motivating factor in the invention of object-oriented approach is to remove some of the flaws encountered in the procedural approach. OOP treats data as a critical element in the program development and does not allow it to flow freely around the system. It ties data more closely to the functions that operate on it and protects it from accidental modification from outside functions.

Object – oriented programming is the most recent concept among programming paradigms. We define “object-oriented programming as an approach that

provides a way of modularize problems by creating partitioned memory area for both data and functions that can be used as templates for creating copies of such modules on demand.

Characteristics of OOP languages

1. **Object**: Objects are entities, which can exist individually. It has its own properties and methods, where properties define the outlook of the object and methods define their procedures.

2. **Class**: It is a template used to define different objects of same type.

3. **Encapsulation**: The data and the methods, which operate on the data, are combined and placed in a group, this phenomenon is known as encapsulation, and the group is known as the object.

4. **Abstraction**: It means hiding of the data of one object of a class from another object of the same class.

5. **Inheritance**: Inheritance is the property by which an existing class can be used to create new classes, by deriving all the properties and methods of the old class to the new class and also adding new properties /methods in the new class. The old class is known as a base class or super class. The new class is known as derived class or sub

class.

6. **Polymorphism**: Polymorphism means “One interface and multiple methods” i.e. one interface can be used to provide different functionalities.

   There are two kinds of polymorphism:

   a) **Compile time polymorphism**: It is also known as early binding, as the interfaces are bind with their methods at compile time. It is accomplished using function overloading and operator overloading.
   b) **Run time polymorphism**: It is also known as late binding. In run time polymorphism the interface and its method bind at the time of execution. It is accomplished using virtual function.

7. Multiple Inheritance: When two or more classes are used to define a single class, then it is known as multiple inheritance.

Benefits of OOP

Object oriented contributes to solution of many problems associated with the development quality of software products. The new technology promises greater programmer productivity, better quality of software and lesser maintenance cost. The principal advantages are:

- Through inheritance, we can eliminate redundant code and extend the use of existing classes.
- We can build programs from the standard working modules that communicate with one another rather than having to start writing the code from scratch. This leads to saving of development time and higher productivity.
- The principle of data hiding helps the programmer to build secure programs that cannot be invaded by code in other parts of the program
- It is possible to have map objects in the problem domain to those in the program.
- It is easy to partition the work in a project based on objects.
- The data centered design approach enables us to capture more details of a model in implemental form,
- Object-oriented systems can be easily upgraded from small to large systems.
- Message passing techniques for communication between objects make the interface descriptions with external systems much simpler.

Software complexity can be easily managed.

What our system provides to the users

As told we have made our project for 7 users.

For all the 7 users our system provides the following features which are briefly listed below

1. Admin – admin will maintain all the functionality of the school Be it maintaining the library , maintaining the banking system , maintaining the students list, staff list , teacher list , class teacher list , class monitor list , leader list.

2. Teacher – doing all the functions of the teacher like changing his/her details, taking exams, giving marks to students, getting payment and so on.

3. Student – performing all the functions of the students, be it giving exams, marking attendance and storing and changing his/her details, paying the fees

4. Class monitor – performing all the functions of the student with some additional functionality

5. Class teacher – performing all the functions of the teacher with some additional functionality

6. Staff – performing all the functions of the staff and storing and changing his/her details, receiving payment etc

7. Leader – performing all the functions of a class monitor with some additional functionality

WORKING AND DESIGN WITH UML DIAGRAMS

(CODE EXPLAINATION)

All the files of our project

admin.cpp class_monitor.cpp leader.cpp mainpro.cpp

admin.h class_monitor.h leader.h Makefile student.cpp

bank.cpp class_teacher.cpp library.cpp modified_calculator.cpp student.h

bank.h class_teacher.h library.h modified_calculator.h teacher.cpp

calculator.cpp entry_detail.h login_system.cpp staff.cpp teacher.h

calculator.h exam_marks.h login_system.h staff.h

We start by creating a object of loginsystem in the main function.

The loginsystem class has a function login through which we can perform the task of logging into the system to perform various functions.

mainpro.cpp

```cpp
#include<bits/stdc++.h>
using namespace std;

#include "login_system.h"

int main(){

    cout << "Welcome Sir/Mam\n"
         << "Please login into your account\n";

    LoginSystem login;

    login.login();

    return 0;

}
```

Login_system class

Login_system.h

```cpp
#ifndef __LOGIN_SYSTEM
#define __LOGIN_SYSTEM

#include <bits/stdc++.h>
using namespace std;

#include "admin.h"
#include "exam_marks.h"

class LoginSystem
```

```
{

private:
    string password[7] = {"1", "2", "3", "4", "5", "6", "7"};

    string username[7] = {"Admin", "Teacher", "Student", "Staff", "Class_teacher", "Class_
monitor", "Leader"};

    string access_rights[7] = {"ADMIN", "TEACHER", "STUDENT", "STAFF", "CLASS_TEACHER", "C
LASS_MONITOR", "LEADER"};

    string username_attempted;

    string password_attempted;

    string choice;

    bool accessgranted;

public:
    admin admin1;

    void login();
};

#endif
```

We first have the login page in which you can have 7 access rights (each access rights has separate username and password)

In order to ensure security we have provided username and passwords to 7 different users which will use the system.

Access Rights

1. ADMIN
2. TEACHER
3. STUDENT
4. STAFF
5. CLASS_TEACHER
6. CLASS_MONITOR
7. LEADER

You will be asked to input username and password

If the username and password matches the stored username and password then they will be directed to the functions of the provided username and passwork

By using username and password verification we have tried to ensure the security of our system so that the person who don't know the username and password cannot enter our system and change and manipulate the data and operate the functions of various users using our system.

If you are a Admin

Then you have type Admin as username and 1 as password.

Similarly, if you are Teacher then username Teacher and password 2.

And so on for other 5 users.

Entering the correct username and password will provide you the respective ACCESS RIGHT through which you can perform your functions and manage your data.

Note we are maintaining different passwords and usernames for different Users. This will ensure that a student would be granted the access right of teacher and teacher wont be given rights of admin and so on.

Explaining the class Loginsystem

Major Fields

Private

String array → password[7] - storing the passwords of different users for different access rights

String array → username[7] -- storing the usernames of different users for different access rights

String array → access_right[7] - storing the access right of different users for different access rights

Username_accepted → storing the username inputted by the user

Password_accepted → storing the password inputted by the user

Bool assesgranted → 0 if access is not granted ,1 if granted

Public

In the public section of the loginsystem we have created an object of class admin

Q Why is admin object in public

We are using creating only 1 instance of the admin class and we will be using this object to run our system. We are using the login function of loginsystem to call the various functions of this admin object so declaring it as public or private are equivalent (hardly makes a difference) because we are using it inside the same class (inside the function login) which can anyways access this object be it public or private.

Methods

Public

Login() to login the user into the system

Explaining the login function

(not the internal implementation but what are the functionality which it will serve)

1. You can login as many times as possible if the username and password (different for different users eg teacher and student have different username and password)
2. The system will allow only 5 consecutive wrong passwords after that it will hold the screen for 30 second (sleep of 30 sec)
3. After the 30 sec you can try to login again
4. And the same process continues

Implementation of the login function

Loginsystem.cpp

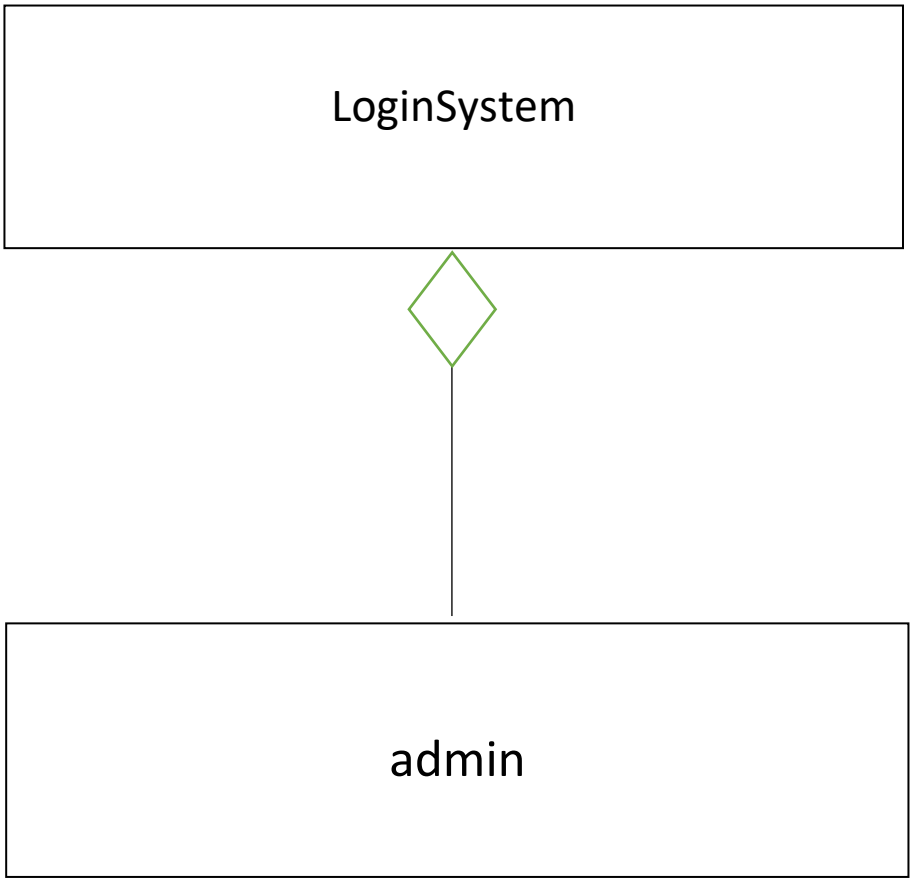
```
#include <bits/stdc++.h>

using namespace std;

#include "login_system.h"

void sleep(float seconds)
```

```
{

    clock_t startClock = clock();

    float secondsAhead = seconds * CLOCKS_PER_SEC;

    // do nothing until the elapsed time has passed.

    while (clock() < startClock + secondsAhead)
        ;

    return;
}

void LoginSystem ::login()
{

    static int login_attempts;

    accessgranted = 0;

    login_attempts++;

    string username_attempted;

    cout << "Enter Your Username:- ";

    cin >> username_attempted;

    cout << "Enter Your Password:- ";

    cin >> password_attempted;

    for (int i = 0; i < 7; i++)
    {

        if (username[i] == username_attempted && password[i] == password_attempted
)

        {

            accessgranted = 1;

            cout << "Account Logged In!\n";

            cout << "ACCESS LEVEL :- " << access_rights[i] << '\n';

            if (access_rights[i] == "ADMIN")

            {
```

```
            //GIVING ACCESS TO ALL FUNCTIONS

            // admin class

            admin1.admin_functions();
        }

        else if (access_rights[i] == "STUDENT")

        {

            //GIVING ACCESS TO STUDENT

            // student class

            admin1.student_functions();
        }

        else if (access_rights[i] == "TEACHER")

        {

            //GIVING ACCESS TO TEACHER FUNCTIONS

            // teacher class

            admin1.teacher_functions();
        }

        else if (access_rights[i] == "STAFF")

        {

            //GIVING ACCESS TO STAFF FUNCTIONS

            // staff class

            admin1.staff_functions();
        }

        else if (access_rights[i] == "CLASS_TEACHER")
        {

            admin1.class_teacher_functions();
        }

        else if (access_rights[i] == "CLASS_MONITOR")
        {

            admin1.class_monitor_functions();
        }
```

```
        else if (access_rights[i] == "LEADER")
        {

            admin1.leader_functions();
        }

        cout << "Wish to log out or leave\n"

             << "Press and enter (LO) for log out or (L) for leave\n";

        string ending;

        cin >> ending;

        if (ending == "LO" || ending == "lo")

            login();

        else

            exit(0);
    }
}

// else

// {

if (accessgranted == 0)
{

    if (login_attempts == 5)

    {

        cout << "Too Many Login Attempts!\n"

             << "Wait for 30 seconds\n";

        sleep(30.0);
    }

    cout << "Invalid Username Attempted!!!\nPlease Try Again!\n";

    sleep(5.0);

    // system("clear"); // check

    login();
}
```

```
    // }
}
```


LoginSystem
admin

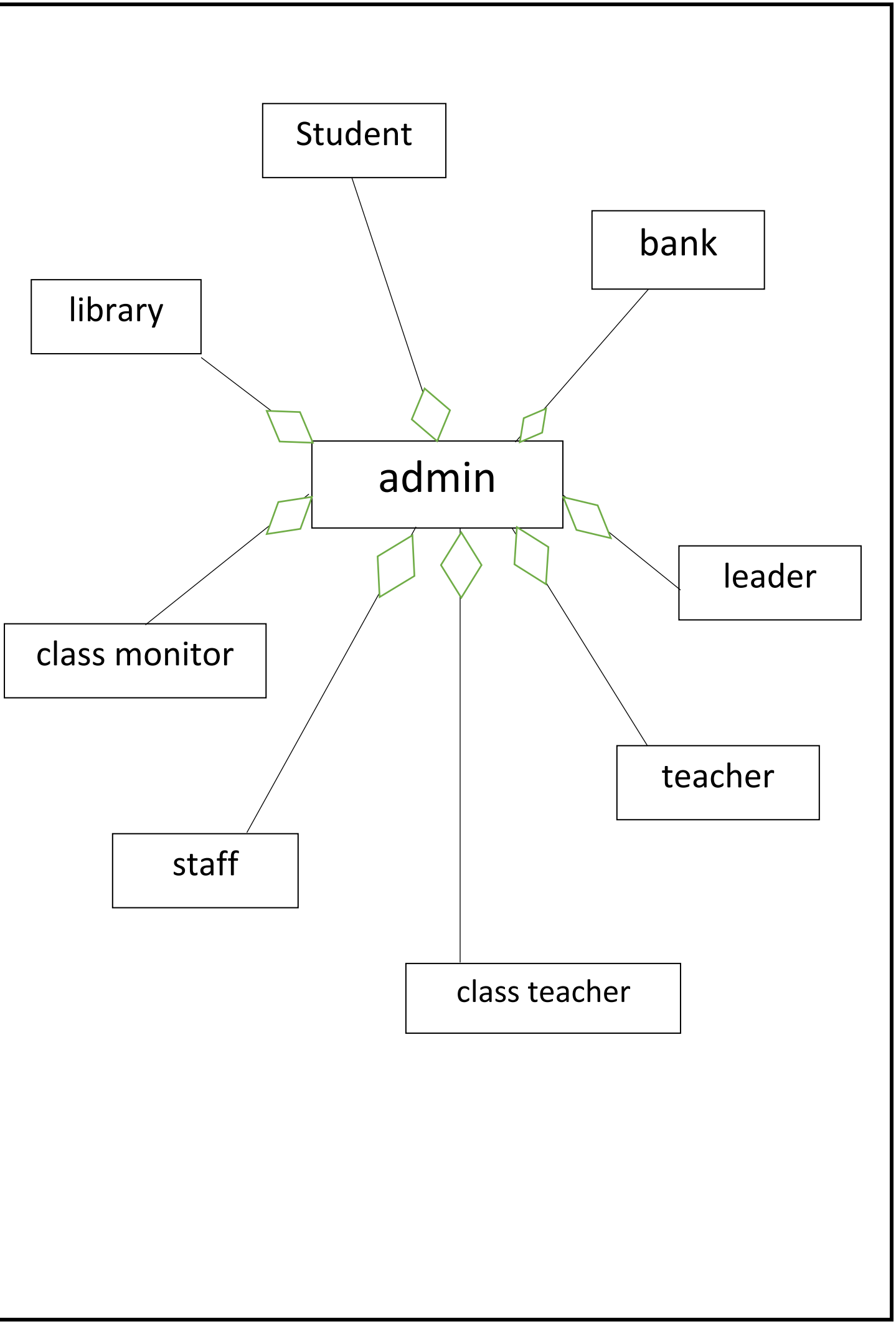
Before explaining the functions of the 7 users we will explain all the classes and there functions

Admin class

admin.h

```
#ifndef __ADMIN
#define __ADMIN

#include <bits/stdc++.h>
using namespace std;

#include "bank.h"
#include "library.h"
#include "leader.h"
#include "class_teacher.h"

#include "exam_marks.h"

class admin
{
private:
    static int student_id;
    static int staff_id;

    bank obj_bank;
    Library obj_library;

    vector<student> student_list;
    vector<teacher> teacher_list;
    vector<staff> staff_list;
    vector<class_monitor> class_monitor_list;
    vector<class_teacher> class_teacher_list;
    vector<leader> leader_list;

public:
    int get_school_balance();

    void admin_functions();
    void student_functions();
    void teacher_functions();
    void staff_functions();

    void class_monitor_functions();
    void class_teacher_functions();
    void leader_functions();
```

```
    void add_student();
    void remove_student();

    void add_class_monitor();
    void remove_class_monitor();

    void add_leader();
    void remove_leader();

    void pay_staff();

    void add_teacher();
    void remove_teacher();

    void add_class_teacher();
    void remove_class_teacher();

    void add_staff();
    void remove_staff();

    void lib_add_book();
    void get_id_book();
};

#endif
```


Student
bank
library
admin
leader
class monitor
teacher
staff
class teacher

bank class

bank.h

```
#ifndef __BANK
#define __BANK

#include <bits/stdc++.h>
using namespace std;

class bank
{
private:
    static int transition_id;

    static int school_balance;

    vector<pair<int, int>> student_transaction; //<student_id ,amount_paid>

    vector<pair<int, int>> staff_transaction; //<staff_id ,amount_paid>

public:

    int get_school_balance();

    void credit(int value);
    void debit(int value);

    void perform_student_transaction(int student_id, int amount);
    void pay_payment_staff(int staff_id, int amount);

    void generate_reciept_student(int, int, int);

    void generate_reciept_staff(int, int, int);

};

#endif
```

Calculator class

Calculator.h

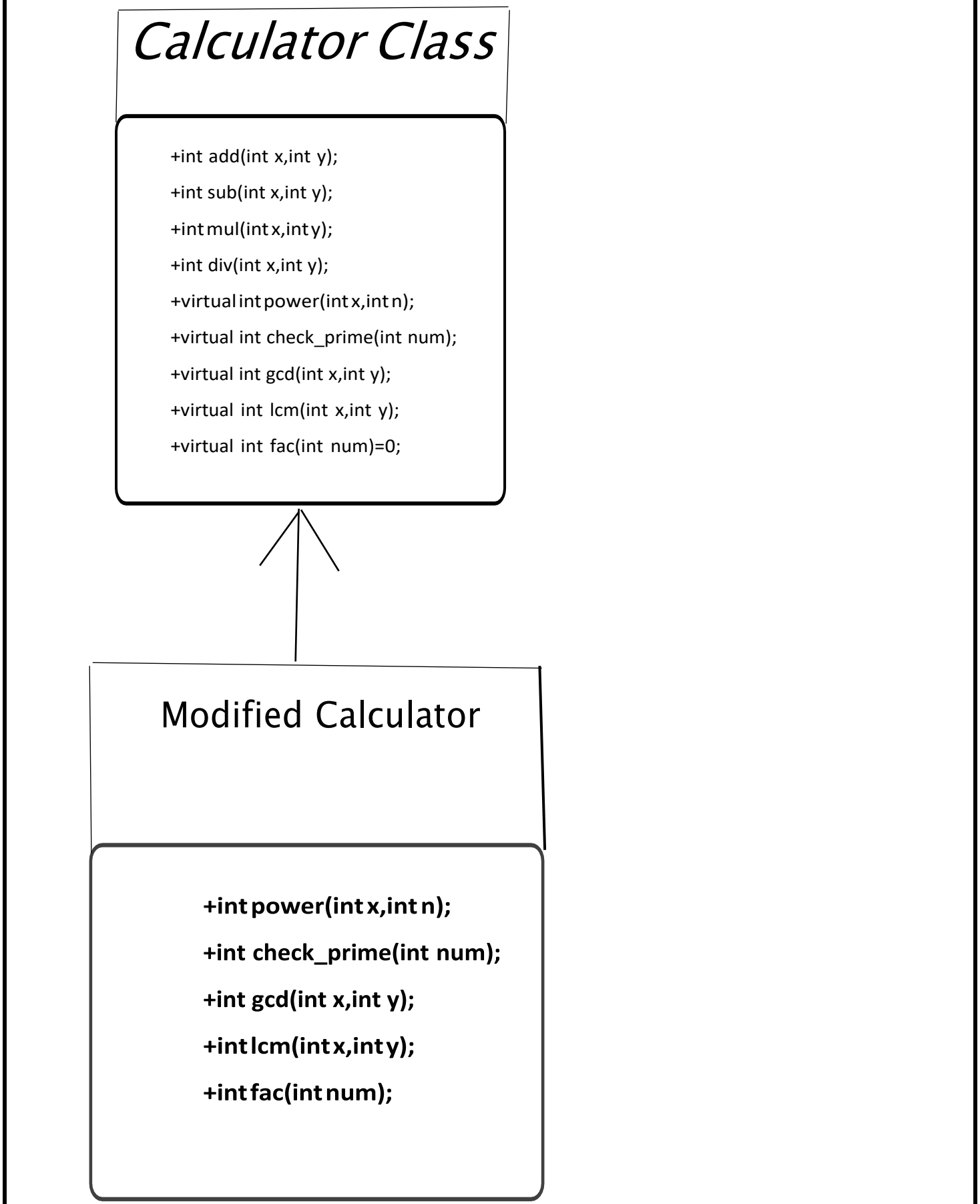
```
#ifndef __CALCULATOR
#define __CALCULATOR

#include <bits/stdc++.h>
using namespace std;
```

```
class calculator
{
public:
    int add(int x, int y);
    int sub(int x, int y);
    int mul(int x, int y);
    int div(int x, int y);

    virtual int power(int x, int n);
    virtual int check_prime(int num); // 1 if prime   0 if not
    virtual int gcd(int x, int y);
    virtual int lcm(int x, int y);

    virtual int fac(int num) = 0;
};

#endif
```

Class Modified_calculator

Modified_calculator.h

```
#ifndef __MODIFIED_CALCULATOR
#define __MODIFIED_CALCULATOR

#include <bits/stdc++.h>
using namespace std;

#include "calculator.h"

class modified_calculator : public calculator
{
public:
    int power(int x, int n);
    int check_prime(int num);
    int gcd(int x, int y);
    int lcm(int x, int y);
    int fac(int num);
};

#endif
```


Calculator Class
+int add(int x,int y);
+int sub(int x,int y);
+int mul(int x,int y);
+int div(int x,int y);
+virtual int power(int x,int n);
+virtual int check_prime(int num);
+virtual int gcd(int x,int y);
+virtual int lcm(int x,int y);
+virtual int fac(int num)=0;
Modified Calculator
+int power(int x,int n);
+int check_prime(int num);
+int gcd(int x,int y);
+int lcm(int x,int y);
+int fac(int num);

Library class

Library.h

```
#ifndef __LIBRARY
#define __LIBRARY

#include <bits/stdc++.h>
using namespace std;

#include "entry_detail.h"

class Library
{
private:
    static int student_book_id;
    static int teacher_book_id;

    vector<book_detail> student_store;
    vector<book_detail> teacher_store;

    map<int, vector<entry_detail>> student_record;
    map<int, vector<entry_detail>> teacher_record;

public:
    void add_book(string, int); // parameter (name of the book, 1 --> stdent/  2 -
-> teacher))

    void add_to_students_store(string);
    void add_to_teachers_store(string);

    int get_id_student(string); // returns the id of the book with given name from
 students store
    int get_id_teacher(string); // returns the id of the book with given name from
 teachers store

    void issue_student(string, string, int); // name of book, date of issue of boo
k,roll no of the student

    void submit_student(int, string, string); // roll_no , book name , date of sub
mission

    void issue_teacher(string, string, int);

    void submit_teacher(int, string, string); // staff id , book name , date of su
bmission

    void display();
};
```

```
#endif
```

Student class

Student.h

```
#ifndef __STUDENT
#define __STUDENT

#include <bits/stdc++.h>
using namespace std;

#include "modified_calculator.h"

#include "exam_marks.h"

class student
{
private:
    string username;
    int roll_no; // unique for all students --> so used as this to identify it
    string name;
    string email;
    string phone_no;
    // int admission_no;
    string date_of_birth;
    int standard;

    int number_of_exams;
    vector<pair<string, int>> attandence;

    vector<exam_marks> exam;

    int percentage;

    int fee_paid; // 1 if paid // 0 if not paid

public:
    void set_username(string username);
    void set_roll_no(int roll_no);
    void set_name(string name);
    void set_email(string email);
    void set_phone_no(string phone_no);
    void set_date_of_birth(string date_of_birth);
```

```
    void set_standard(int standard);
    void update_standard(int status); // 1 for pass   0 for fail

    void add_date_for_attandence(string data, int present); // date, 1 or 0    1 f
or present 0 for absent

    string get_username();
    int get_roll_no();
    string get_name();
    string get_email();
    string get_phone_no();
    string get_date_of_birth();
    int get_standard();

    void give_exam(); // just write exam

    void give_maths_exam(); // calculator use

    vector<pair<string, int>> get_attendance_list();

    void show_report(); // report card

    void set_percentage();

    float get_percentage();

    // fee paid
    void set_fee_paid(int check); // 1 if paid // 0 if not paid
    int get_fee_paid();

    void add_exam_marks(exam_marks m);

    void display();
};

#endif
```

Class_monitor class

Class_monitor.h

```
#ifndef __CLASS_MONITOR
#define __CLASS_MONITOR

#include <bits/stdc++.h>
using namespace std;
```

```
#include "student.h"

#include "exam_marks.h"

class class_monitor : public student
{
private:
public:
    void maintain_the_class();

    void display();
};

#endif
```

Leader class

Leader.h

```
#ifndef __LEADER
#define __LEADER

#include <bits/stdc++.h>
using namespace std;

#include "class_monitor.h"

#include "exam_marks.h"

class leader : public class_monitor
{
public:
    void maintain_the_monitor();
    void display();
};

#endif
```

STUDENT

- string Username
- string Name
- int Roll Number
- string EmailID
- string PhoneNo
- string DateOfBirth
- int standard
- vector<pair<string,int>>attendance
- vector<exam_marks>Exam;
- int percentage
- int feePaid

+void set_roll_no(int roll_no);

+void set_name(string name);

+void set_email(string email);

+void set_phone_no(string phone_no);

+void set_date_of_birth(string date_of_birth);

+void set_standard(int standard);

+void update_standard(int status);

+void add_date_for_attandence(string data, int present)

+string get_username();

+int get_roll_no();

+string get_name();

+string get_email();

AND SOME MORE FUNCTIONS

Pointing to the student class of the previous page

Class Monitor's Class

+void maintain_the_class();

+void display();

Leader Class

+void maintain_the_monitor();

+void display();

Class staff

Staff.h

```
#ifndef __STAFF
#define __STAFF

#include <bits/stdc++.h>
using namespace std;

class staff
{
private:
    string username;
    int staff_id; // unique for all students --> so used as this to identify it
    string designation;
    string name;
    string email;
    string address;
    string phone_no;
    string date_of_birth;
    int salary;

    vector<pair<string, int>> attandence;

    int payment_done; // 1 if payment is done , 0 otherwise not done

public:
    void set_username(string username); // used to set as well as upadte // therefore took sperately all
    void set_staff_id(int user_id);
    void set_designation(string designation);
    void set_name(string name);
    void set_email(string email);
    void set_phone_no(string phone_no);
    void set_date_of_birth(string date_of_birth);

    void set_salary(int salary);

    void set_address(string address);

    string get_address();

    void add_date_for_attandence(string data, int present); // date, 1 or 0    1 for present 0 for absent
```

```
    string get_username();
    int get_staff_id();
    string get_designation();
    string get_name();
    string get_email();
    string get_phone_no();
    string get_date_of_birth();
    int get_salary();

    vector<pair<string, int>> get_attendance_list();

    void set_payment_done(int check); // 1 if done // 0 if not done
    int get_payment_done();

    void display();
};

#endif
```

Class teacher

Teacher.h

```
#ifndef __TEACHER
#define __TEACHER

#include <bits/stdc++.h>
using namespace std;

#include "staff.h"
#include "modified_calculator.h"

class teacher : public staff
{
private:
    vector<pair<int, string>> subject; // batch , subject

public:
    void add_subject(int batch_no, string subject_name);
    void conduct_exam();
```

```
    void get_subject();

    void use_calculator();

    void display();
};

#endif
```

Class Class_teacher

Class_teacher.h

```
#ifndef __CLASS_TEACHER
#define __CLASS_TEACHER

#include <bits/stdc++.h>
using namespace std;

#include "teacher.h"

#include "exam_marks.h"

class class_teacher : public teacher
{
private:
    int batch_no;

public:
    void set_batch_no(int batch_no);
    int get_batch_no();

    exam_marks give_marks();

    void display();
};

#endif
```

STAFF

-string username;

-int staff_id;

-string designation;

-string name;

-string email;

-string address;

-string phone_no;

-string date_of_birth;

-int salary;

-vector <pair <string ,int > > attandence;

-int payment_done;

+void set_username(string username);

+void set_staff_id(int user_id);

+void set_designation(string designation);

+void set_name(string name);

+void set_email(string email);

+void set_phone_no(string phone_no);

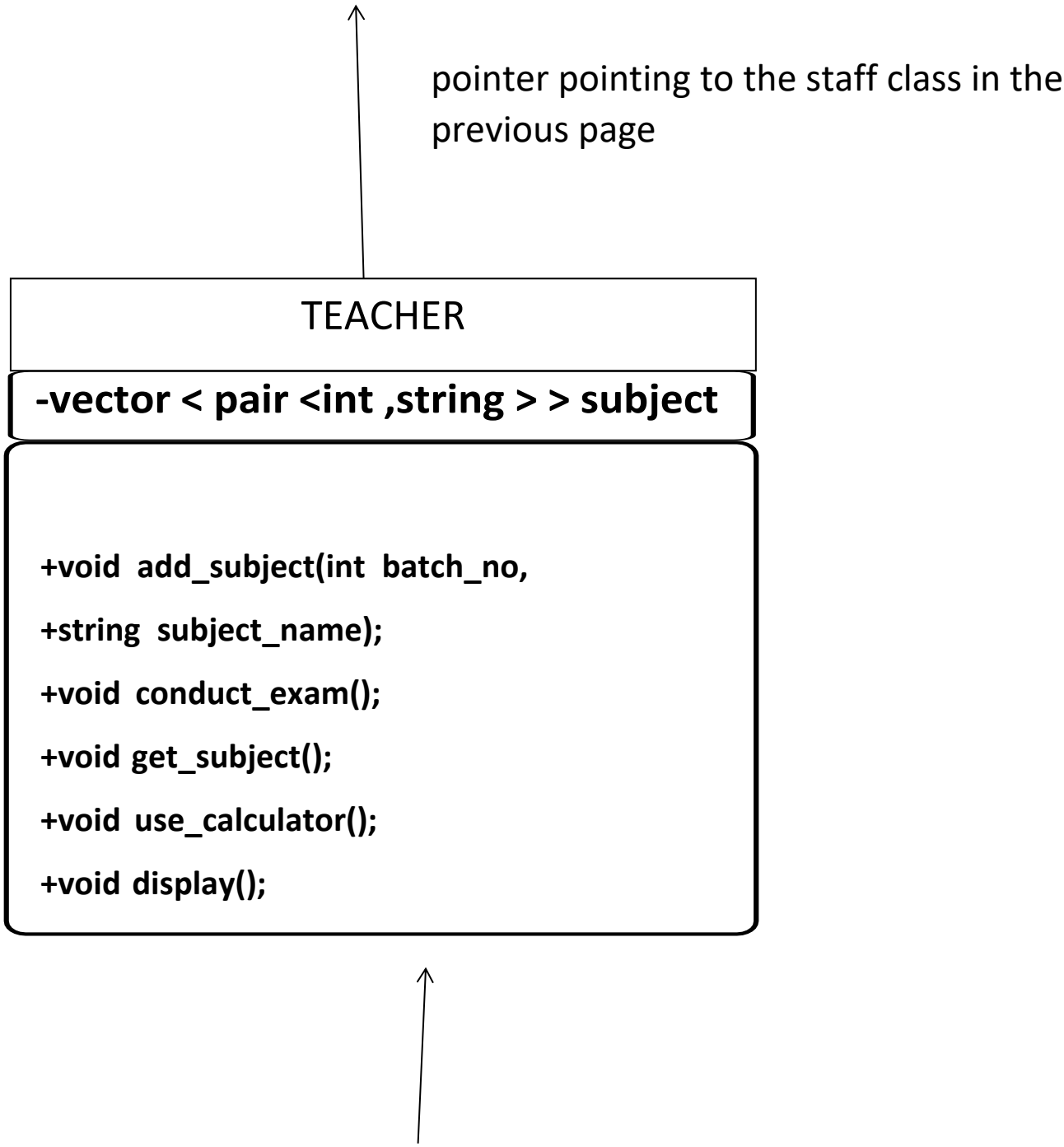
+void set_date_of_birth(string date_of_birth);

+void set_salary(int salary);

+void set_address(string address);

+string get_address();

and some more functions


pointer pointing to the staff class in the previous page
TEACHER
-vector < pair <int ,string > > subject
+void add_subject(int batch_no,
+string subject_name);
+void conduct_exam();
+void get_subject();
+void use_calculator();
+void display();
Class teacher Class
-int batch_no;
+void set_batch_no(int batch_no);
+int get_batch_no();
+exam_marks give_marks();
+ void display();

As informed earlier the object of the admin class will perform all the functions

1. It will perform its own functions(we will explain them in detail)
2. It will provide a system which will allow all the different users to use there functions

So we have moved from the loginsystem class to the admin class

admin class

admin.h

```
#ifndef __ADMIN
#define __ADMIN

#include <bits/stdc++.h>

using namespace std;

#include "bank.h"

#include "library.h"

#include "leader.h"

#include "class_teacher.h"

#include "exam_marks.h"

class admin
{

private:
    static int student_id;

    static int staff_id;

    bank obj_bank;
```

```
    Library obj_library;

    vector<student> student_list;

    vector<teacher> teacher_list;

    vector<staff> staff_list;

    vector<class_monitor> class_monitor_list;

    vector<class_teacher> class_teacher_list;

    vector<leader> leader_list;

public:
    int get_school_balance();

    void admin_functions();

    void student_functions();

    void teacher_functions();

    void staff_functions();

    void class_monitor_functions();

    void class_teacher_functions();

    void leader_functions();

    void add_student();

    void remove_student();

    void add_class_monitor();

    void remove_class_monitor();

    void add_leader();

    void remove_leader();

    void pay_staff();

    void add_teacher();

    void remove_teacher();

    void add_class_teacher();
```

```
    void remove_class_teacher();

    void add_staff();

    void remove_staff();

    void lib_add_book();

    void get_id_book();
};

#endif
```

In the implementation part of the login function of the loginsystem class we were inputting the username and password

If the username and password were of admin

ACCESS RIGHT – ADMIN

Then we will call the admin_functions() using the object of admin we created

From admin.cpp → admin_functions()

```
void admin::admin_functions()
{

    cout << "Welcome Admin here!! How can I help you  " << endl;

    cout << "MENU" << endl;

    cout << "-----------------------" << endl;

    int choice;

    cout << "Press 1 to add student " << endl;

    cout << "Press 2 to add teacher " << endl;

    cout << "Press 3 to add staff " << endl;

    cout << "Press 4 to remove student " << endl;
```

```
    cout << "Press 5 to remove teacher " << endl;

    cout << "Press 6 to remove staff " << endl;

    cout << "Press 7 to add class monitor (student) " << endl;

    cout << "Press 8 to add class teacher (teacher) " << endl;

    cout << "Press 9 to add leader (student) " << endl;

    cout << "Press 10 to remove class monitor (student) " << endl;

    cout << "Press 11 to remove class teacher (teacher) " << endl;

    cout << "Press 12 to remove leader (student) " << endl;

    cout << "Press 13 to pay the staff" << endl;

    cout << "Press 14 to get the school balance" << endl;

    cout << "Press 15 to add book" << endl;

    cout << "Press 16 to check id of book" << endl;

    cout << "Press 17 to display student list " << endl;

    cout << "Press 18 to display class monitor list " << endl;

    cout << "Press 19 to display leader list" << endl;

    cout << "Press 20 to display staff list " << endl;

    cout << "Press 21 to display teacher list " << endl;

    cout << "Press 22 to display class teacher list " << endl;

    cout << "Press 23 to display library contents " << endl;

    cout << "Press 24 to exit" << endl;

    cout << "Enter your choice " << endl;

    cin >> choice;

    if (choice == 1)
    {

        add_student();
    }
```

```
    else if (choice == 2)
    {

        add_teacher();
    }

    else if (choice == 3)
    {

        add_staff();
    }

    else if (choice == 4)
    {

        remove_student();
    }

    else if (choice == 5)
    {

        remove_teacher();
    }

    else if (choice == 6)
    {

        remove_staff();
    }

    else if (choice == 7)
    {

        add_class_monitor();
    }

    else if (choice == 8)
    {

        add_class_teacher();
    }

    else if (choice == 9)
    {

        add_leader();
    }

    else if (choice == 10)
    {
```

```
        remove_class_monitor();
    }

    else if (choice == 11)
    {

        remove_class_teacher();
    }

    else if (choice == 12)
    {

        remove_leader();
    }

    else if (choice == 13)
    {

        pay_staff();
    }

    else if (choice == 14)
    {

        int x = obj_bank.get_school_balance();

        cout << x << endl;
    }

    else if (choice == 15)
    {

        lib_add_book();
    }

    else if (choice == 16)
    {

        get_id_book();
    }

    else if (choice == 17)
    {

        int size = student_list.size();

        cout << "List size :- " << size << endl;

        for (int i = 0; i < size; i++)
        {
```

```
            cout << "Student number " << i + 1 << endl;

            student_list[i].display();
        }
    }

    else if (choice == 18)
    {

        int size = class_monitor_list.size();

        cout << "List size :- " << size << endl;

        for (int i = 0; i < size; i++)
        {

            cout << "Class monitor number " << i + 1 << endl;

            class_monitor_list[i].display();
        }
    }

    else if (choice == 19)
    {

        int size = leader_list.size();

        cout << "List size :- " << size << endl;

        for (int i = 0; i < size; i++)
        {

            cout << "Leader number " << i + 1 << endl;

            leader_list[i].display();
        }
    }

    else if (choice == 20)
    {

        int size = staff_list.size();

        cout << "List size :- " << size << endl;

        for (int i = 0; i < size; i++)
        {

            cout << "Staff number " << i + 1 << endl;

            staff_list[i].display();
```

```
        }
    }

    else if (choice == 21)
    {

        int size = teacher_list.size();

        cout << "List size :- " << size << endl;

        for (int i = 0; i < size; i++)
        {

            cout << "teacher number " << i + 1 << endl;

            teacher_list[i].display();
        }
    }

    else if (choice == 22)
    {

        int size = class_teacher_list.size();

        cout << "List size :- " << size << endl;

        for (int i = 0; i < size; i++)
        {

            cout << "class_teacher number " << i + 1 << endl;

            class_teacher_list[i].display();
        }
    }

    else if (choice == 23)
    {

        obj_library.display();
    }

    else if (choice == 24)
    {

        return;
    }

    cout << "Do you want to perform more functions < y/Y for yes > < n/N for no >"
;

    char ch;
```

```
    cin >> ch;

    if (ch == 'y' || ch == 'Y')
    {

        admin_functions();
    }

    else
    {

        return;
    }
}
```

You will be shown the menu list of all the functions that the admin can perform

You can enter your choice accordingly to perform the function you want the admin to do

Explaining the functions that the admin can perform

First choice → Add student

Real life relation → student comes to the admin and says add me in the system and the admin adds the student into the system by asking the details of the student

From admin.cpp → add_student()

```
void admin::add_student()
{
    student obj1;

    obj1.set_fee_paid(0);

    string str;
    cout << "Enter Student name :- ";
    cin >> str;
    obj1.set_name(str);
```

```
    string username;
    cout << "Enter Username :- ";
    cin >> username;
    obj1.set_username(username);

    student_id++;
    obj1.set_roll_no(student_id);

    string email;
    cout << "Enter Email :-";
    cin >> email;
    obj1.set_email(email);

    string phone_no;
    cout << "Enter Phone No :-";
    cin >> phone_no;
    obj1.set_phone_no(phone_no);

    string date_of_birth;
    cout << "Enter Date Of Birth :-";
    cin >> date_of_birth;
    obj1.set_date_of_birth(date_of_birth);

    int standard;
    cout << "Enter Standard :-";
    cin >> standard;
    obj1.set_standard(standard);

    student_list.push_back(obj1);

    cout << "Student has been added successfully!" << endl;
}
```

We create an student object and store the student object in the vector of student_list (which is field of admin).

Now this student is added in our system and now he/she can perform his/her functions using this object of student created.

Note that the admin gives the student a unique id (roll number in case of student) which the student has to remember in order to perform any of his/her operation

When we will see the student_functions () it will ask your id(roll no) and depending on the id you give it will first validate you and then will allow you to operate your functions

Third choice→ remove student

If you want to remove a student from your system then you can remove a student

From admin.cpp → remove_student()

```
void admin:: remove_student(){
    int id;
    cout<<"Enter the roll number (student id) to be removed ";
    cin>>id;

    int check=0;

    int size=student_list.size();

    auto it=student_list.begin();
    for(int i=0;i<size;i++){
        if (student_list[i].get_roll_no()==id) {
            student_list.erase(it);
            check=1;
            break;
        }
        it++;
    }

    if (check){
        cout<<"The student with id "<<id<<" has been delted successfully "<<endl;
    }
    else {
        cout <<"No such student exist "<<endl;
    }

}
```

It will ask the id (here roll no) of the student to be removed

If id matches it will remove the student otherwise not

We do the similar thing with 6 other users so adding and removing them, in all 12 choices so the first 12 choices are for adding and removing the users from the system

Choice 13

We know the admin manages the bank also using the object of the bank created in the admin class → we manage the bank

So 1 function of school is to pay the employees / staff

So the admin has the function to pay the fees of the staff

Working

It will first ask the staff id which was generated at the time of adding staff to the system

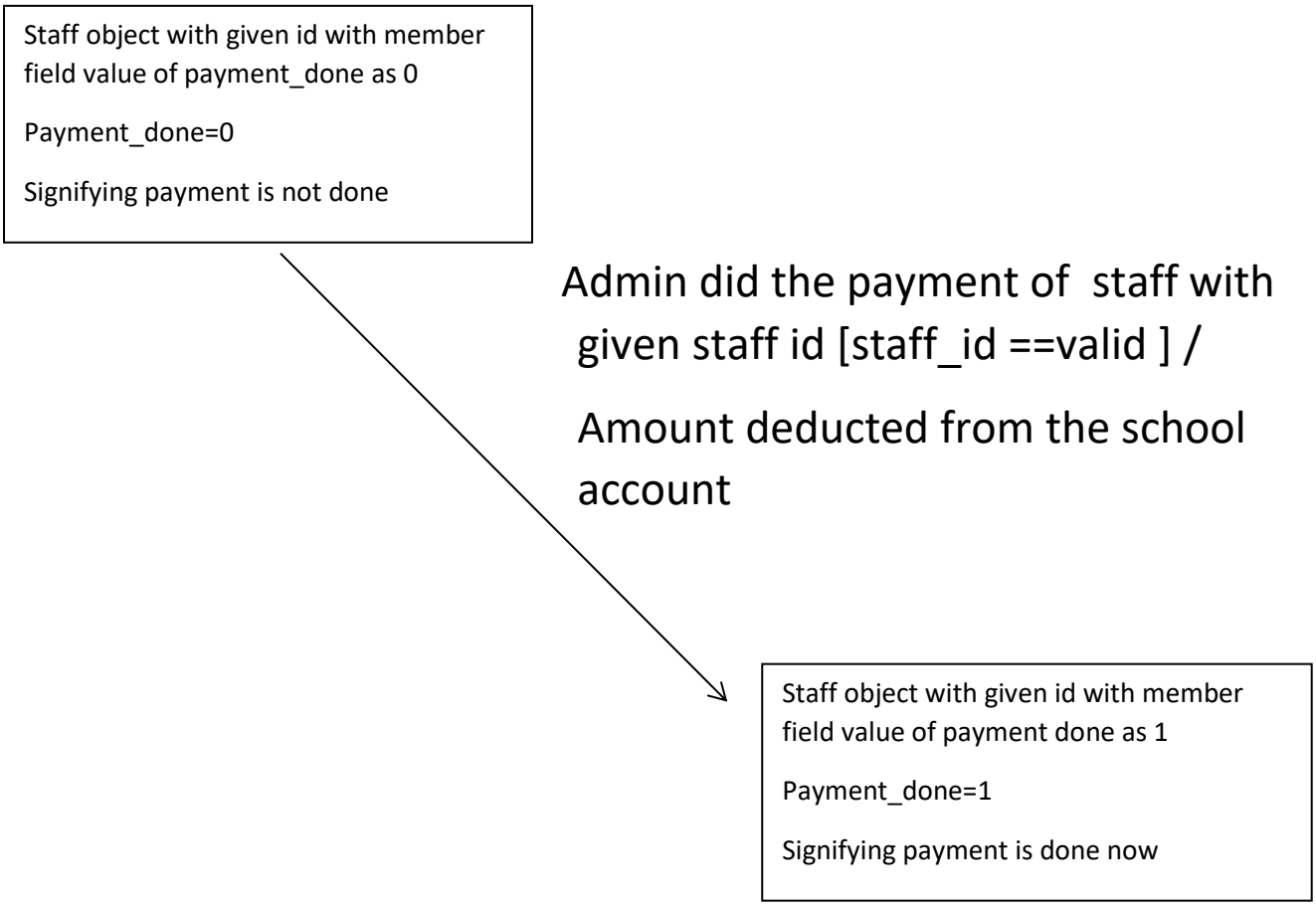
(you will input the id)

It will check in its database that this staff exists or not

(Note that before paying the fees of the staff you should have that staff registered in the system (added in the system) Otherwise it will show no staff with this id is present in our system)

If the id is not valid then the admin will not pay the fees and the will exit from this choice

Otherwise the following state change takes place



Choice 14

Get the current school balance from the bank (initially 100000)

Choice 15

The admin will add books to the library

Working →

The admin will ask the book name and ask if the book is for teachers or for the students

Depending on the input the admin through the object of the library created in the admin class

Add these books to the respective stores → student store or teacher store

(Note that the library object you created will give unique ids to the books you are adding in the system which you can see using the choice 15)

Choice 15

Through the object of library created in the admin you can fetch the id of the books you have added in the library

Choices 16 to 23 are to display the list of various fields in the admin class

(The choices and their respective task is shown in the menu)

Choice 24 is to exit the admin_functions() function

ACCESS RIGHT – STUDENT

If you are student then though a function named students_function() you will be able to perform your operations

From admin.cpp → student_functions()

```
void admin::student_functions()
{

    int id;
    cout << "Enter your student id ";
    cin >> id;

    int check = 0;

    int size = student_list.size();

    student *ptr;

    auto it = student_list.begin();
    for (int i = 0; i < size; i++)
    {
        if (student_list[i].get_roll_no() == id)
        {
            ptr = &student_list[i];
            check = 1;
            break;
        }
        it++;
    }

    if (check == 0)
    {
        cout << "The student with this id do not exist " << endl;
    }
    else
    {

        while (1)
        {
```

```
            cout << "Student with this id " << id << " here !!" << endl;
            cout << "MENU " << endl;
            cout << "--------------------------------" << endl;
            cout << "Press 1 for change username " << endl;
            cout << "Press 2 for change name " << endl;
            cout << "Press 3 for change email " << endl;
            cout << "Press 4 for change phone_no " << endl;
            cout << "Press 5 for change date_of_birth " << endl;
            cout << "Press 6 for change standard " << endl;
            cout << "Press 7 for get username " << endl;
            cout << "Press 8 for get name " << endl;
            cout << "Press 9 for get email " << endl;
            cout << "Press 10 for get phone_no " << endl;
            cout << "Press 11 for get date_of_birth " << endl;
            cout << "Press 12 for get standard " << endl;
            cout << "Press 13 for get roll no " << endl;
            cout << "Press 14 for add date for attendance " << endl;
            // cout << "Press 15 for marking todays attendance " << endl;
            cout << "Press 16 to give exam" << endl;
            cout << "Press 17 to get the percentage " << endl;
            cout << "Press 18 to get the attendance list " << endl;
            cout << "Press 19 to get the report-card " << endl;

            // new // bank
            cout << "Press 20 to pay the fees" << endl;
            cout << "Press 21 to get fee paid or not" << endl;

            // new // lib
            // check if you like want the lib infro in student class as well or no
t
            cout << "Press 22 to issue a book " << endl;
            cout << "Press 23 to submit a book" << endl;

            cout << "Press 24 to display" << endl;
            cout << "Press 25 to give maths exam" << endl;

            cout << "Press 26 to exit" << endl;

            int choice;
            cout << "Enter your choice " << endl;
            cin >> choice;

            if (choice == 1)
            {
                string username;
                cout << "Enter Username :- ";
                cin >> username;
                ptr->set_username(username);
            }
            else if (choice == 2)
```

```
        {
            string str;
            cout << "Enter Student name :- ";
            cin >> str;
            ptr->set_name(str);
        }
        else if (choice == 3)
        {
            string email;
            cout << "Enter Email :-";
            cin >> email;
            ptr->set_email(email);
        }
        else if (choice == 4)
        {
            string phone_no;
            cout << "Enter Phone No :-";
            cin >> phone_no;
            ptr->set_phone_no(phone_no);
        }
        else if (choice == 5)
        {
            string date_of_birth;
            cout << "Enter Date Of Birth :-";
            cin >> date_of_birth;
            ptr->set_date_of_birth(date_of_birth);
        }
        else if (choice == 6)
        {
            int standard;
            cout << "Enter Standard :-";
            cin >> standard;
            ptr->set_standard(standard);
        }

        else if (choice == 7)
        {
            string str = ptr->get_username();
            cout << "Username -" << str << endl;
        }
        else if (choice == 8)
        {
            string str = ptr->get_name();
            cout << "Name -" << str << endl;
        }
        else if (choice == 9)
        {
            string str = ptr->get_email();
            cout << "Email -" << str << endl;
        }
        else if (choice == 10)
```

```
        {
            string str = ptr->get_phone_no();
            cout << "Phone number -" << str << endl;
        }
        else if (choice == 11)
        {
            string str = ptr->get_date_of_birth();
            cout << "Date of birth -" << str << endl;
        }
        else if (choice == 12)
        {
            int str = ptr->get_standard();
            cout << "Standard -" << str << endl;
        }
        else if (choice == 13)
        {
            int str = ptr->get_roll_no();
            cout << "Roll no -" << str << endl;
        }
        else if (choice == 14)
        {
            string str;
            int x;
            cout << "Enter the date " << endl;
            cin >> str;
            cout << "Enter <1,0> ( 1 for present and 0 for absent ) " << endl;
            cin >> x;
            ptr->add_date_for_attandence(str, x);
        }
        else if (choice == 15)
        {
            ptr->mark_todays_attandence();
        }
        else if (choice == 16)
        {
            ptr->give_exam();
        }
        else if (choice == 17)
        {
            float x = ptr->get_percentage();
            cout << "Percentage - " << x << endl;
        }
        else if (choice == 18)
        {
            vector<pair<string, int>> v;
            v = ptr->get_attendance_list();

            cout << "Date"
                 << "        "
                 << "Status" << endl;
            for (auto x : v)
```

```
            {
                cout << x.first << "  " << x.second << endl;
            }
        }
        else if (choice == 19)
        {
            ptr->show_report();
        }

        else if (choice == 20)
        {
            // fix this fees somehow say 1000 store it somewhere
            obj_bank.perform_student_transaction(ptr->get_roll_no(), 1000);
            ptr->set_fee_paid(1);
        }

        else if (choice == 21)
        {
            int x = ptr->get_fee_paid();
            cout << x << endl;
        }

        else if (choice == 22)
        {
            string book;
            cout << "Enter book name you want " << endl;
            cin >> book;
            string date;
            // input or use prabhav function to fetch todays date
            // now we are inputing
            cout << "Enter the todays date " << endl;
            cin >> date;
            obj_library.issue_student(book, date, id);
        }

        else if (choice == 23)
        {
            string book;
            cout << "Enter book name you want to return " << endl;
            cin >> book;
            string date;
            // input or use prabhav function to fetch todays date
            // now we are inputing
            cout << "Enter the todays date ( return date ) " << endl;
            cin >> date;
            obj_library.submit_student(id, book, date);
        }

        else if (choice == 24)
        {
            ptr->display();
```

```
            }
            else if (choice == 25)
            {
                ptr->give_maths_exam();
            }
            else if (choice == 26)
            {
                return;
            }

            cout << "Do you want to perform more functions < y/Y for yes > < n/N f
or no >";
            char ch;
            cin >> ch;

            if (ch == 'y' || ch == 'Y')
            {
                // teacher_functions();
                continue;
            }
            else
            {
                // return;
                break;
            }

        } // while loop

    } // else condition

} // complete function
```

Working →

The system will ask you to input your id (roll no)

You will input the id

The system will validate you

If a student with that id matches then you will be displayed the menu screen through which you can perform your functions

Choices 1 to 16 are just setting and getting functions through which you can get and set the values of various fields of the student

They will go the things as stated in the menu

Choice 17

This will show the percentage of the student

This percentage is initially set to be 0

Also note that the student cannot change the marks he/she got, neither he can change the percentage he/she got. All the marks and percentage are initialised as 0 and only the class teacher of the respective batch can give marks to student(in turn give the percentage to the student because we calculate the percentage from the marks only)

We will discuss the working of this when we will discuss the functions of class teacher

Choice 18 for the attendance list

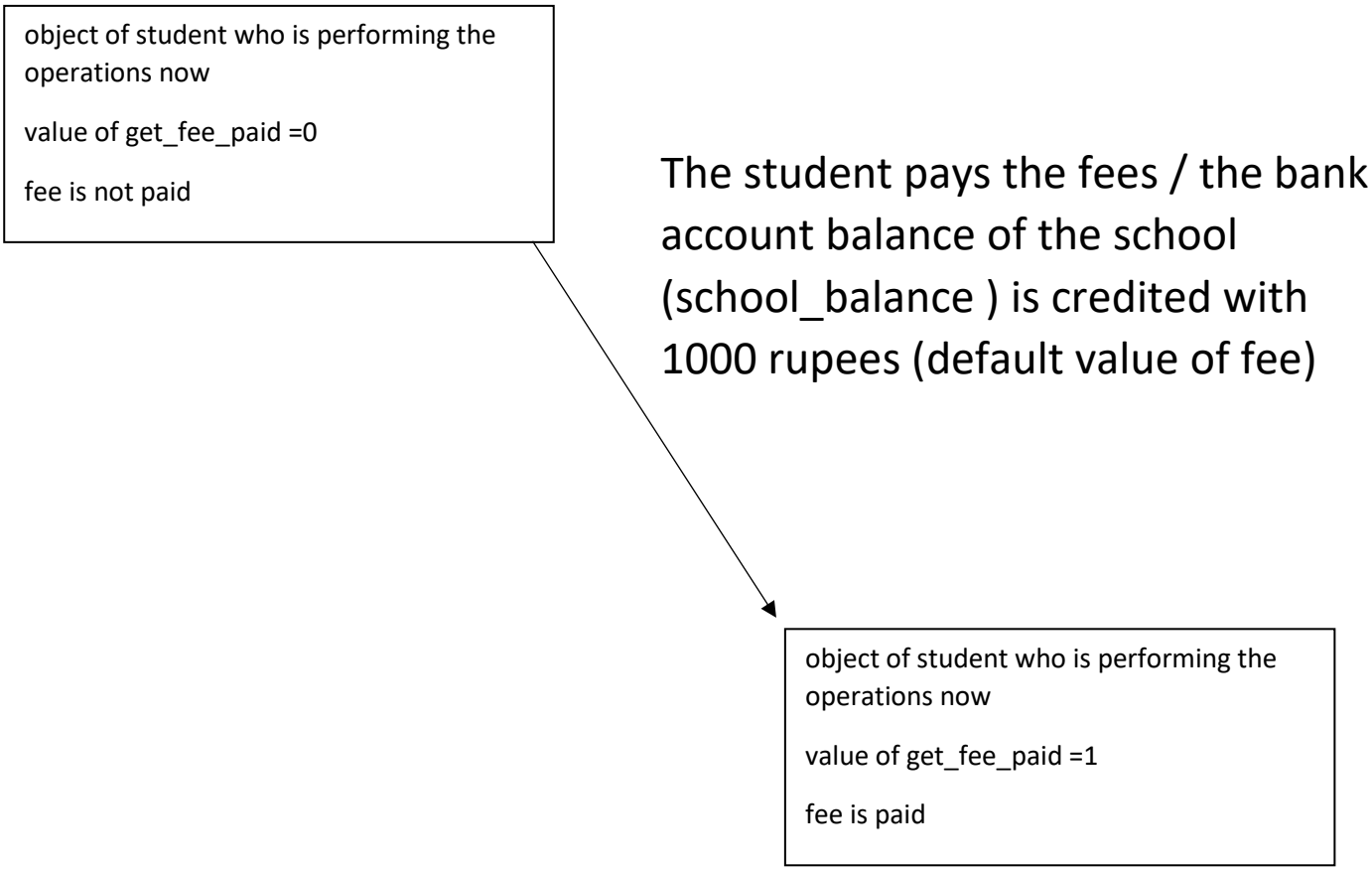
Choice 19 for report card

Choice 20 to pay the fees

Choice 21 to get the value of get_fee_paid field in the student class which will be 0 by-default signifying that the student has not paid the fees

If you call choice 21 initially then it will show 0 → fee not paid

Then if you call choice 20 it will pay the fees of the student and the following state transition take



Interacting with the library

Choice 22

Issuing the book

Choice 23

Submitting the book

The object of the library is maintaining all the records of book issued / submitted etc which can be displayed using the admin function

Choice 24 to display major fields of this student object

Choice 25 to give maths exam

From student.cpp → give_maths_exam()

```
void student::give_maths_exam()
{

    calculator *ptr;
    modified_calculator obj;
    ptr = &obj;
```

```
    cout << "Welcome " << endl;
    cout << "--------------------" << endl;
    cout << "Menu" << endl;
    cout << "--------------------" << endl;
    cout << "Press 1 to add" << endl;
    cout << "Press 2 to subtract" << endl;
    cout << "Press 3 to multiply" << endl;
    cout << "Press 4 to divide" << endl;
    cout << "Press 5 to find power (x^n)" << endl;
    cout << "Press 6 to check if a number is prime or not " << endl;
    cout << "Press 7 to find gcd of two numbers" << endl;
    cout << "Press 8 to find lcm of two numbers" << endl;
    cout << "Press 9 to find the factorial of a number" << endl;
    cout << "Press 10 to exit " << endl;

    int choice;
    cout << "Enter your choice" << endl;
    cin >> choice;

    if (choice == 1)
    {
        int x, y;
        cout << "Enter two numbers" << endl;
        cin >> x >> y;
        int ans = ptr->add(x, y);
        cout << ans << endl;
    }
    else if (choice == 2)
    {
        int x, y;
        cout << "Enter two numbers" << endl;
        cin >> x >> y;
        int ans = ptr->sub(x, y);
        cout << ans << endl;
    }
    else if (choice == 3)
    {
        int x, y;
        cout << "Enter two numbers" << endl;
        cin >> x >> y;
        int ans = ptr->mul(x, y);
        cout << ans << endl;
    }
    else if (choice == 4)
    {
        int x, y;
        cout << "Enter two numbers" << endl;
        cin >> x >> y;
        int ans = ptr->div(x, y);
        cout << ans << endl;
    }
```

```
    else if (choice == 5)
    {
        int x, n;
        cout << "Enter two numbers" << endl;
        cin >> x >> n;
        int ans = ptr->power(x, n);
        cout << ans << endl;
    }
    else if (choice == 6)
    {
        int x;
        cout << "Enter a number" << endl;
        cin >> x;
        int ans = ptr->check_prime(x);
        cout << ans << endl;
    }
    else if (choice == 7)
    {
        int x, y;
        cout << "Enter two numbers" << endl;
        cin >> x >> y;
        int ans = ptr->gcd(x, y);
        cout << ans << endl;
    }
    else if (choice == 8)
    {
        int x, y;
        cout << "Enter two numbers" << endl;
        cin >> x >> y;
        int ans = ptr->lcm(x, y);
        cout << ans << endl;
    }

    else if (choice == 9)
    {
        int num;
        cout << "Enter the number ";
        cin >> num;
        int ans = ptr->fac(num);
        cout << ans << endl;
    }

    else if (choice == 10)
    {
        return;
    }

    cout << "Do you want to perform more functions < y/Y for yes > < n/N for no >"
;
    char ch;
    cin >> ch;
```

```
    if (ch == 'y' || ch == 'Y')
    {
        give_maths_exam();
    }
    else
    {
        return;
    }
}
```

Please note that we are using Polymorphism here

Calculator is pure abstract class (as shown in the class diagram of calculator) and Modified_calculator is derived class.

Here we have made pointer of base class pointing to the object of derived class and using derived class functions to give the output because of virtual and pure virtual functions.

Choice 26 to exit

ACCESS RIGHT → CLASS_MONITOR

From admin.cpp → class_monitor_functions()

```
void admin::class_monitor_functions()
{

    int id;
    cout << "Enter your student id ";
    cin >> id;

    int check = 0;

    int size = class_monitor_list.size();

    class_monitor *ptr;

    auto it = class_monitor_list.begin();
    for (int i = 0; i < size; i++)
    {
        if (class_monitor_list[i].get_roll_no() == id)
        {
            ptr = &class_monitor_list[i];
            check = 1;
            break;
        }
        it++;
    }

    if (check == 0)
    {
        cout << "The student with this id do not exist " << endl;
    }
    else
    {

        while (1)
        {

            cout << "The class monitor (student) with this id" << id << " here" <<
 endl;
            cout << "MENU " << endl;
            cout << "--------------------------------" << endl;
            cout << "Press 1 for change username " << endl;
            cout << "Press 2 for change name " << endl;
            cout << "Press 3 for change email " << endl;
            cout << "Press 4 for change phone_no " << endl;
            cout << "Press 5 for change date_of_birth " << endl;
            cout << "Press 6 for change standard " << endl;
```

```
            cout << "Press 7 for get username " << endl;
            cout << "Press 8 for get name " << endl;
            cout << "Press 9 for get email " << endl;
            cout << "Press 10 for get phone_no " << endl;
            cout << "Press 11 for get date_of_birth " << endl;
            cout << "Press 12 for get standard " << endl;
            cout << "Press 13 for get roll no " << endl;
            cout << "Press 14 for add date for attendance " << endl;
            cout << "Press 15 for marking todays attendance " << endl;
            cout << "Press 16 to give exam" << endl;
            cout << "Press 17 to get the percentage " << endl;
            cout << "Press 18 to get the attendance list " << endl;
            cout << "Press 19 to get the report-card " << endl;

            // unique here
            // cout<<"Press 20 to change standard you are monitoring"<<endl;
            // cout<<"Press 21 to get the standard you are monitoring"<<endl;
            cout << "Press 22 to maintain the class" << endl;
            // new
            cout << "Press 23 to pay the fees" << endl;
            cout << "Press 24 to get fee paid or not" << endl;

            // new // lib
            // check if you like want the lib infro in student class as well or no
t
            cout << "Press 25 to issue a book " << endl;
            cout << "Press 26 to submit a book" << endl;

            cout << "Press 27 to display" << endl;
            cout << "Press 28 to exit" << endl;

            int choice;
            cout << "Enter your choice " << endl;
            cin >> choice;

            if (choice == 1)
            {
                string username;
                cout << "Enter Username :- ";
                cin >> username;
                ptr->set_username(username);
            }
            else if (choice == 2)
            {
                string str;
                cout << "Enter Student name :- ";
                cin >> str;
                ptr->set_name(str);
            }
            else if (choice == 3)
            {
```

```
            string email;
            cout << "Enter Email :-";
            cin >> email;
            ptr->set_email(email);
        }
        else if (choice == 4)
        {
            string phone_no;
            cout << "Enter Phone No :-";
            cin >> phone_no;
            ptr->set_phone_no(phone_no);
        }
        else if (choice == 5)
        {
            string date_of_birth;
            cout << "Enter Date Of Birth :-";
            cin >> date_of_birth;
            ptr->set_date_of_birth(date_of_birth);
        }
        else if (choice == 6)
        {
            int standard;
            cout << "Enter Standard :-";
            cin >> standard;
            ptr->set_standard(standard);
        }

        else if (choice == 7)
        {
            string str = ptr->get_username();
            cout << "Username -" << str << endl;
        }
        else if (choice == 8)
        {
            string str = ptr->get_name();
            cout << "Name -" << str << endl;
        }
        else if (choice == 9)
        {
            string str = ptr->get_email();
            cout << "Email -" << str << endl;
        }
        else if (choice == 10)
        {
            string str = ptr->get_phone_no();
            cout << "Phone number -" << str << endl;
        }
        else if (choice == 11)
        {
            string str = ptr->get_date_of_birth();
            cout << "Date of birth -" << str << endl;
```

```
        }
        else if (choice == 12)
        {
            int str = ptr->get_standard();
            cout << "Standard -" << str << endl;
        }
        else if (choice == 13)
        {
            int str = ptr->get_roll_no();
            cout << "Roll no -" << str << endl;
        }
        else if (choice == 14)
        {
            string str;
            int x;
            cout << "Enter the date " << endl;
            cin >> str;
            cout << "Enter <1,0> ( 1 for present and 0 for absent ) " << endl;
            cin >> x;
            ptr->add_date_for_attandence(str, x);
        }
        else if (choice == 15)
        {
            ptr->mark_todays_attandence();
        }
        else if (choice == 16)
        {
            ptr->give_exam();
        }
        else if (choice == 17)
        {
            float x = ptr->get_percentage();
            cout << "Percentage - " << x << endl;
        }
        else if (choice == 18)
        {
            vector<pair<string, int>> v;
            v = ptr->get_attendance_list();

            cout << "Date"
                 << "         "
                 << "Status" << endl;
            for (auto x : v)
            {
                cout << x.first << "   " << x.second << endl;
            }
        }
        else if (choice == 19)
        {
            ptr->show_report();
        }
```

```
            // else if (choice==20) {
            //     int x;
            //     cout<<"Enter the standard you are monitoring"<<endl;
            //     cin>>x;
            //     ptr->set_standard(x);
            // }
            // else if (choice==21) {
            //     int x=ptr->get_standard();
            //     cout<<"Standard you are monitoring "<<x<<endl;
            // }
            else if (choice == 22)
            {
                ptr->maintain_the_class();
            }

            else if (choice == 23)
            {
                // fix this fees somehow say 1000 store it somewhere
                obj_bank.perform_student_transaction(ptr->get_roll_no(), 1000);
                ptr->set_fee_paid(1);
            }

            else if (choice == 24)
            {
                int x = ptr->get_fee_paid();
                cout << x << endl;
            }

            else if (choice == 25)
            {
                string book;
                cout << "Enter book name you want " << endl;
                cin >> book;
                string date;
                // input or use prabhav function to fetch todays date
                // now we are inputing
                cout << "Enter the todays date " << endl;
                cin >> date;
                obj_library.issue_student(book, date, id);
            }

            else if (choice == 26)
            {
                string book;
                cout << "Enter book name you want to return " << endl;
                cin >> book;
                string date;
                // input or use prabhav function to fetch todays date
                // now we are inputing
                cout << "Enter the todays date ( return date ) " << endl;
                cin >> date;
```

```
                obj_library.issue_student(book, date, id);
            }

            else if (choice == 27)
            {
                ptr->display();
            }
            else if (choice == 28)
            {
                return;
            }

            cout << "Do you want to perform more functions < y/Y for yes > < n/N f
or no >";
            char ch;
            cin >> ch;

            if (ch == 'y' || ch == 'Y')
            {
                // teacher_functions();
                continue;
            }
            else
            {
                // return;
                break;
            }

        } // while loop

    } // else condition

} // complete function
```

All the choices remain same as in students with additional

choice 22 - of maintaining the class

ACCESS LEVEL – LEADER

From admin.cpp → leader_functions()

```
void admin::leader_functions()
{
    int id;
    cout << "Enter your student id ";
    cin >> id;

    int check = 0;

    int size = leader_list.size();

    leader *ptr;

    auto it = leader_list.begin();
    for (int i = 0; i < size; i++)
    {
        if (leader_list[i].get_roll_no() == id)
        {
            ptr = &leader_list[i];
            check = 1;
            break;
        }
        it++;
    }

    if (check == 0)
    {
        cout << "The student with this id do not exist " << endl;
    }
    else
    {

        while (1)
        {

            cout << "The leader student with this id " << id << " here" << endl;
            cout << "MENU " << endl;
            cout << "--------------------------------" << endl;
            cout << "Press 1 for change username " << endl;
            cout << "Press 2 for change name " << endl;
            cout << "Press 3 for change email " << endl;
            cout << "Press 4 for change phone_no " << endl;
            cout << "Press 5 for change date_of_birth " << endl;
            cout << "Press 6 for change standard " << endl;
            cout << "Press 7 for get username " << endl;
            cout << "Press 8 for get name " << endl;
            cout << "Press 9 for get email " << endl;
```

```
            cout << "Press 10 for get phone_no " << endl;
            cout << "Press 11 for get date_of_birth " << endl;
            cout << "Press 12 for get standard " << endl;
            cout << "Press 13 for get roll no " << endl;
            cout << "Press 14 for add date for attendance " << endl;
            cout << "Press 15 for marking todays attendance " << endl;
            cout << "Press 16 to give exam" << endl;
            cout << "Press 17 to get the percentage " << endl;
            cout << "Press 18 to get the attendance list " << endl;
            cout << "Press 19 to get the report-card " << endl;

            // unique here
            cout << "Press 20 to change standard you are monitoring" << endl;
            cout << "Press 21 to get the standard you are monitoring" << endl;
            cout << "Press 22 to maintain the class" << endl;

            cout << "Press 23 to maintain the monitors " << endl;

            // new
            cout << "Press 24 to pay the fees" << endl;
            cout << "Press 25 to get fee paid or not" << endl;

            // new // lib
            // check if you like want the lib infro in student class as well or no
t
            cout << "Press 26 to issue a book " << endl;
            cout << "Press 27 to submit a book" << endl;

            cout << "Press 28 to display" << endl;
            cout << "Press 29 to exit" << endl;

            int choice;
            cout << "Enter your choice " << endl;
            cin >> choice;

            if (choice == 1)
            {
                string username;
                cout << "Enter Username :- ";
                cin >> username;
                ptr->set_username(username);
            }
            else if (choice == 2)
            {
                string str;
                cout << "Enter Student name :- ";
                cin >> str;
                ptr->set_name(str);
            }
            else if (choice == 3)
            {
```

```
            string email;
            cout << "Enter Email :-";
            cin >> email;
            ptr->set_email(email);
        }
        else if (choice == 4)
        {
            string phone_no;
            cout << "Enter Phone No :-";
            cin >> phone_no;
            ptr->set_phone_no(phone_no);
        }
        else if (choice == 5)
        {
            string date_of_birth;
            cout << "Enter Date Of Birth :-";
            cin >> date_of_birth;
            ptr->set_date_of_birth(date_of_birth);
        }
        else if (choice == 6)
        {
            int standard;
            cout << "Enter Standard :-";
            cin >> standard;
            ptr->set_standard(standard);
        }

        else if (choice == 7)
        {
            string str = ptr->get_username();
            cout << "Username -" << str << endl;
        }
        else if (choice == 8)
        {
            string str = ptr->get_name();
            cout << "Name -" << str << endl;
        }
        else if (choice == 9)
        {
            string str = ptr->get_email();
            cout << "Email -" << str << endl;
        }
        else if (choice == 10)
        {
            string str = ptr->get_phone_no();
            cout << "Phone number -" << str << endl;
        }
        else if (choice == 11)
        {
            string str = ptr->get_date_of_birth();
            cout << "Date of birth -" << str << endl;
```

```
        }
        else if (choice == 12)
        {
            int str = ptr->get_standard();
            cout << "Standard -" << str << endl;
        }
        else if (choice == 13)
        {
            int str = ptr->get_roll_no();
            cout << "Roll no -" << str << endl;
        }
        else if (choice == 14)
        {
            string str;
            int x;
            cout << "Enter the date " << endl;
            cin >> str;
            cout << "Enter <1,0> ( 1 for present and 0 for absent ) " << endl;
            cin >> x;
            ptr->add_date_for_attandence(str, x);
        }
        else if (choice == 15)
        {
            ptr->mark_todays_attandence();
        }
        else if (choice == 16)
        {
            ptr->give_exam();
        }
        else if (choice == 17)
        {
            float x = ptr->get_percentage();
            cout << "Percentage - " << x << endl;
        }
        else if (choice == 18)
        {
            vector<pair<string, int>> v;
            v = ptr->get_attendance_list();

            cout << "Date"
                 << "         "
                 << "Status" << endl;
            for (auto x : v)
            {
                cout << x.first << "   " << x.second << endl;
            }
        }
        else if (choice == 19)
        {
            ptr->show_report();
        }
```

```
        else if (choice == 20)
        {
            int x;
            cout << "Enter the standard you are monitoring" << endl;
            cin >> x;
            ptr->set_standard(x);
        }
        else if (choice == 21)
        {
            int x = ptr->get_standard();
            cout << "Standard you are monitoring " << x << endl;
        }
        else if (choice == 22)
        {
            ptr->maintain_the_class();
        }

        // unique
        else if (choice == 23)
        {
            ptr->maintain_the_monitor();
        }

        else if (choice == 24)
        {
            // fix this fees somehow say 1000 store it somewhere
            obj_bank.perform_student_transaction(ptr->get_roll_no(), 1000);
            ptr->set_fee_paid(1);
        }

        else if (choice == 25)
        {
            int x = ptr->get_fee_paid();
            cout << x << endl;
        }

        else if (choice == 26)
        {
            string book;
            cout << "Enter book name you want " << endl;
            cin >> book;
            string date;
            // input or use prabhav function to fetch todays date
            // now we are inputing
            cout << "Enter the todays date " << endl;
            cin >> date;
            obj_library.issue_student(book, date, id);
        }

        else if (choice == 27)
```

```
            {
                string book;
                cout << "Enter book name you want to return " << endl;
                cin >> book;
                string date;
                // input or use prabhav function to fetch todays date
                // now we are inputing
                cout << "Enter the todays date ( return date ) " << endl;
                cin >> date;
                obj_library.issue_student(book, date, id);
            }

            else if (choice == 28)
            {
                ptr->display();
            }
            else if (choice == 29)
            {
                return;
            }

            cout << "Do you want to perform more functions < y/Y for yes > < n/N f
or no >";
            char ch;
            cin >> ch;

            if (ch == 'y' || ch == 'Y')
            {
                // teacher_functions();
                continue;
            }
            else
            {
                // return;
                break;
            }

        } // while loop

    } // else condition

} // complete function
```

All the choices remain same as in class_monitor with additional choice 23 of maintaining the class monitors

ACCESS LEVEL → STAFF

Working →

The system will ask you to input your staff id

You will input the id

The system will validate you

If a staff with that id matches then you will be displayed the menu screen through which you can perform your functions

```
void admin::staff_functions()
{
    int id;
    cout << "Enter your staff id ";
    cin >> id;

    int check = 0;

    int size = staff_list.size();

    staff *ptr;

    auto it = staff_list.begin();
    for (int i = 0; i < size; i++)
    {
        if (staff_list[i].get_staff_id() == id)
        {
            ptr = &staff_list[i];
            check = 1;
            break;
        }
        it++;
    }

    if (check == 0)
    {
        cout << "The staff with this id do not exist " << endl;
    }
    else
```

```
{

    // designation change
    // address change
    // salary change

    while (1)
    {

        cout << "The staff with this " << id << " here " << endl;
        cout << "MENU " << endl;
        cout << "--------------------------------" << endl;
        cout << "Press 1 for change username " << endl;
        cout << "Press 2 for change name " << endl;
        cout << "Press 3 for change email " << endl;
        cout << "Press 4 for change phone_no " << endl;
        cout << "Press 5 for change date_of_birth " << endl;

        cout << "Press 6 for change designation " << endl;

        cout << "Press 7 for get username " << endl;
        cout << "Press 8 for get name " << endl;
        cout << "Press 9 for get email " << endl;
        cout << "Press 10 for get phone_no " << endl;
        cout << "Press 11 for get date_of_birth " << endl;

        cout << "Press 12 for get designation " << endl;

        cout << "Press 13 for get staff id " << endl;

        cout << "Press 14 for add date for attendance " << endl;
        cout << "Press 15 for marking todays attendance " << endl;

        cout << "Press 16 to set address" << endl;
        cout << "Press 17 to get address" << endl;

        cout << "Press 18 to set salary" << endl;
        cout << "Press 19 to get salary " << endl;

        cout << "Press 20 to get the attendance list " << endl;
        // cout<<"Press 19 to get the report-card "<<endl;

        // new

        cout << "Press 21 to check payment is done or not " << endl;

        cout << "Press 22 to display " << endl;

        cout << "Press 23 to exit " << endl;

        int choice;
```

```
        cout << "Enter your choice " << endl;
        cin >> choice;

        if (choice == 1)
        {
            string username;
            cout << "Enter Username :- ";
            cin >> username;
            ptr->set_username(username);
        }
        else if (choice == 2)
        {
            string str;
            cout << "Enter Staff name :- ";
            cin >> str;
            ptr->set_name(str);
        }
        else if (choice == 3)
        {
            string email;
            cout << "Enter Email :-";
            cin >> email;
            ptr->set_email(email);
        }
        else if (choice == 4)
        {
            string phone_no;
            cout << "Enter Phone No :-";
            cin >> phone_no;
            ptr->set_phone_no(phone_no);
        }
        else if (choice == 5)
        {
            string date_of_birth;
            cout << "Enter Date Of Birth :-";
            cin >> date_of_birth;
            ptr->set_date_of_birth(date_of_birth);
        }
        else if (choice == 6)
        {
            string designation;
            cout << "Enter designation :-";
            cin >> designation;
            ptr->set_designation(designation);
        }

        else if (choice == 7)
        {
            string str = ptr->get_username();
            cout << "Username -" << str << endl;
        }
```

```
        else if (choice == 8)
        {
            string str = ptr->get_name();
            cout << "Name -" << str << endl;
        }
        else if (choice == 9)
        {
            string str = ptr->get_email();
            cout << "Email -" << str << endl;
        }
        else if (choice == 10)
        {
            string str = ptr->get_phone_no();
            cout << "Phone number -" << str << endl;
        }
        else if (choice == 11)
        {
            string str = ptr->get_date_of_birth();
            cout << "Date of birth -" << str << endl;
        }
        else if (choice == 12)
        {
            string str = ptr->get_designation();
            cout << "Designation -" << str << endl;
        }
        else if (choice == 13)
        {
            int x = ptr->get_staff_id();
            cout << "Staff id -" << x << endl;
        }

        else if (choice == 14)
        {
            string str;
            int x;
            cout << "Enter the date " << endl;
            cin >> str;
            cout << "Enter <1,0> ( 1 for present and 0 for absent ) " << endl;
            cin >> x;
            ptr->add_date_for_attandence(str, x);
        }
        else if (choice == 15)
        {
            ptr->mark_todays_attandence();
        }
        else if (choice == 16)
        {
            string str;
            cout << "Enter Address " << endl;
            cin >> str;
            ptr->set_address(str);
```

```
        }
        else if (choice == 17)
        {
            string str = ptr->get_address();
            cout << str << endl;
        }
        else if (choice == 18)
        {
            int x;
            cout << "Enter salary" << endl;
            cin >> x;
            ptr->set_salary(x);
        }
        else if (choice == 19)
        {
            int x = ptr->get_salary();
            cout << "salary  " << x << endl;
        }
        else if (choice == 20)
        {
            vector<pair<string, int>> v;
            v = ptr->get_attendance_list();

            cout << "Date"
                 << "        "
                 << "Status" << endl;
            for (auto x : v)
            {
                cout << x.first << "  " << x.second << endl;
            }
        }
        // else if (choice==19) {
        //     ptr->show_report();
        // }

        else if (choice == 21)
        {
            int x = ptr->get_payment_done();
            cout << x << endl;
        }

        else if (choice == 22)
        {
            ptr->display();
        }

        else if (choice == 23)
        {
            return;
        }
```

```
            cout << "Do you want to perform more functions < y/Y for yes > < n/N f
or no >";
            char ch;
            cin >> ch;

            if (ch == 'y' || ch == 'Y')
            {
                // teacher_functions();
                continue;
            }
            else
            {
                // return;
                break;
            }

        } // while loop

    } // else condition

} // complete function
```

Choice number 1 to 13 for changing and showing the fields in the staff

Choice 14 and 15

For marking attendance

Choice 16 to 19 for changing and showing the fields in the staff

Choice 20 for showing the attendance list

Choice 21 to check payment

We have already shown this in the admin functions how the state of the staff is changed when the admin makes the payment for the staff

Choice 22 to display

Choice 23 to exit

ACCESS LEVEL – TEACHER

From admin.cpp → teacher_functions()

```
void admin::teacher_functions()
{
    int id;
    cout << "Enter your staff id ";
    cin >> id;

    int check = 0;

    int size = teacher_list.size();

    teacher *ptr;

    auto it = teacher_list.begin();
    for (int i = 0; i < size; i++)
    {
        if (teacher_list[i].get_staff_id() == id)
        {
            ptr = &teacher_list[i];
            check = 1;
            break;
        }
        it++;
    }

    if (check == 0)
    {
        cout << "The staff with this id do not exist " << endl;
    }
    else
```

```
{

    // designation change
    // address change
    // salary change

    while (1)
    {

        cout << "The teacher with this id " << id << " here !!" << endl;
        cout << "MENU " << endl;

        cout << "--------------------------------" << endl;
        cout << "Press 1 for change username " << endl;
        cout << "Press 2 for change name " << endl;
        cout << "Press 3 for change email " << endl;
        cout << "Press 4 for change phone_no " << endl;
        cout << "Press 5 for change date_of_birth " << endl;

        cout << "Press 6 for change designation " << endl;

        cout << "Press 7 for get username " << endl;
        cout << "Press 8 for get name " << endl;
        cout << "Press 9 for get email " << endl;
        cout << "Press 10 for get phone_no " << endl;
        cout << "Press 11 for get date_of_birth " << endl;

        cout << "Press 12 for get designation " << endl;

        cout << "Press 13 for get staff id " << endl;

        cout << "Press 14 for add date for attendance " << endl;
        cout << "Press 15 for marking todays attendance " << endl;

        cout << "Press 16 to set address" << endl;
        cout << "Press 17 to get address" << endl;

        cout << "Press 18 to set salary" << endl;
        cout << "Press 19 to get salary " << endl;

        cout << "Press 20 to get the attendance list " << endl;
        // cout<<"Press 19 to get the report-card "<<endl;

        // unique

        cout << "Press 21 to add subject " << endl;

        cout << "Press 22 to get the subject list" << endl;

        cout << "Press 23 to conduct exam " << endl;
```

```
        cout << "Press 24 to check payment is done or not " << endl;

        cout << "Press 25 to display " << endl;

        cout << "Press 26 to exit " << endl;

        // // giving marks to student
        // cout<<"Press 25 to give marks to student "<<endl;

        int choice;
        cout << "Enter your choice " << endl;
        cin >> choice;

        if (choice == 1)
        {
            string username;
            cout << "Enter Username :- ";
            cin >> username;
            ptr->set_username(username);
        }
        else if (choice == 2)
        {
            string str;
            cout << "Enter Staff name :- ";
            cin >> str;
            ptr->set_name(str);
        }
        else if (choice == 3)
        {
            string email;
            cout << "Enter Email :-";
            cin >> email;
            ptr->set_email(email);
        }
        else if (choice == 4)
        {
            string phone_no;
            cout << "Enter Phone No :-";
            cin >> phone_no;
            ptr->set_phone_no(phone_no);
        }
        else if (choice == 5)
        {
            string date_of_birth;
            cout << "Enter Date Of Birth :-";
            cin >> date_of_birth;
            ptr->set_date_of_birth(date_of_birth);
        }
        else if (choice == 6)
        {
            string designation;
```

```
            cout << "Enter designation :-";
            cin >> designation;
            ptr->set_designation(designation);
        }

        else if (choice == 7)
        {
            string str = ptr->get_username();
            cout << "Username -" << str << endl;
        }
        else if (choice == 8)
        {
            string str = ptr->get_name();
            cout << "Name -" << str << endl;
        }
        else if (choice == 9)
        {
            string str = ptr->get_email();
            cout << "Email -" << str << endl;
        }
        else if (choice == 10)
        {
            string str = ptr->get_phone_no();
            cout << "Phone number -" << str << endl;
        }
        else if (choice == 11)
        {
            string str = ptr->get_date_of_birth();
            cout << "Date of birth -" << str << endl;
        }
        else if (choice == 12)
        {
            string str = ptr->get_designation();
            cout << "Designation -" << str << endl;
        }
        else if (choice == 13)
        {
            int x = ptr->get_staff_id();
            cout << "Staff id -" << x << endl;
        }

        else if (choice == 14)
        {
            string str;
            int x;
            cout << "Enter the date " << endl;
            cin >> str;
            cout << "Enter <1,0> ( 1 for present and 0 for absent ) " << endl;
            cin >> x;
            ptr->add_date_for_attandence(str, x);
        }
```

```
        else if (choice == 15)
        {
            ptr->mark_todays_attandence();
        }
        else if (choice == 16)
        {
            string str;
            cout << "Enter Address " << endl;
            cin >> str;
            ptr->set_address(str);
        }
        else if (choice == 17)
        {
            string str = ptr->get_address();
            cout << str << endl;
        }
        else if (choice == 18)
        {
            int x;
            cout << "Enter salary" << endl;
            cin >> x;
            ptr->set_salary(x);
        }
        else if (choice == 19)
        {
            int x = ptr->get_salary();
            cout << "salary  " << x << endl;
        }
        else if (choice == 20)
        {
            vector<pair<string, int>> v;
            v = ptr->get_attendance_list();

            cout << "Date"
                 << "        "
                 << "Status" << endl;
            for (auto x : v)
            {
                cout << x.first << "  " << x.second << endl;
            }
        }
        else if (choice == 21)
        {
            int x;
            string str;
            cout << "Enter the subject ";
            cin >> str;

            cout << "Enter the batch ";
            cin >> x;
            ptr->add_subject(x, str);
```

```
            }

            else if (choice == 22)
            {
                cout << "Subject list is as follows " << endl;
                ptr->get_subject();
            }

            else if (choice == 23)
            {
                ptr->conduct_exam();
            }

            // else if (choice==19) {
            //     ptr->show_report();
            // }

            else if (choice == 24)
            {
                int x = ptr->get_payment_done();
                cout << x << endl;
            }

            else if (choice == 25)
            {
                ptr->display();
            }

            else if (choice == 26)
            {
                return;
            }

            cout << "Do you want to perform more functions < y/Y for yes > < n/N f
or no >";
            char ch;
            cin >> ch;

            if (ch == 'y' || ch == 'Y')
            {
                // teacher_functions();
                continue;
            }
            else
            {
                // return;
                break;
            }

        } // while loop
```

```
    } // else condition

} // complete function
```

All the choices and working remains the same

With 3 additional choices

Choice 21 → add subject (which the teacher is teaching)

Choice 22 → show the subject list (which the teacher is teaching)

Choice 23 → conduct exam

ACCESS LEVEL – CLASS TEACHER

From admin.cpp → class_teacher_functions()

```
void admin::class_teacher_functions()
{

    int id;
    cout << "Enter your staff id ";
    cin >> id;

    int check = 0;

    int size = class_teacher_list.size();

    class_teacher *ptr;

    auto it = class_teacher_list.begin();
    for (int i = 0; i < size; i++)
    {
        if (class_teacher_list[i].get_staff_id() == id)
        {
            ptr = &class_teacher_list[i];
            check = 1;
```

```
            break;
        }
        it++;
    }

    if (check == 0)
    {
        cout << "The staff with this id do not exist " << endl;
        return;
    }
    else
    {

        while (1)
        {

            cout << "MENU " << endl;
            cout << "class Teacher here id -> " << id << endl;
            cout << "--------------------------------" << endl;
            cout << "Press 1 for change username " << endl;
            cout << "Press 2 for change name " << endl;
            cout << "Press 3 for change email " << endl;
            cout << "Press 4 for change phone_no " << endl;
            cout << "Press 5 for change date_of_birth " << endl;

            cout << "Press 6 for change designation " << endl;

            cout << "Press 7 for get username " << endl;
            cout << "Press 8 for get name " << endl;
            cout << "Press 9 for get email " << endl;
            cout << "Press 10 for get phone_no " << endl;
            cout << "Press 11 for get date_of_birth " << endl;

            cout << "Press 12 for get designation " << endl;

            cout << "Press 13 for get staff id " << endl;

            cout << "Press 14 for add date for attendance " << endl;
            cout << "Press 15 for marking todays attendance " << endl;

            cout << "Press 16 to set address" << endl;
            cout << "Press 17 to get address" << endl;

            cout << "Press 18 to set salary" << endl;
            cout << "Press 19 to get salary " << endl;

            cout << "Press 20 to get the attendance list " << endl;


            cout << "Press 21 to add subject " << endl;
```

```
            cout << "Press 22 to get the subject list" << endl;

            cout << "Press 23 to conduct exam " << endl;

            cout << "Press 24 to change the branch " << endl;
            cout << "Press 25 to get the branch " << endl;

            cout << "Press 26 to check payment is done or not " << endl;

            cout << "Press 27 to issue a book " << endl;

            cout << "Press 28 to submit a book" << endl;

            // giving marks to student  --> check later
            cout << "Press 29 to give marks to student " << endl;
            // all the teachers gives the marks to the class teacher and class tea
cher gives them to student

            cout << "Press 30 to change student entries " << endl; // being the cl
ass teacher --> it can

            cout << "Press 31 to display " << endl;

            cout << "Press 32 to exit " << endl;

            int choice;
            cout << "Enter your choice " << endl;
            cin >> choice;

            if (choice == 1)
            {
                string username;
                cout << "Enter Username :- ";
                cin >> username;
                ptr->set_username(username);
            }
            else if (choice == 2)
            {
                string str;
                cout << "Enter Staff name :- ";
                cin >> str;
                ptr->set_name(str);
            }
            else if (choice == 3)
            {
                string email;
                cout << "Enter Email :-";
                cin >> email;
                ptr->set_email(email);
            }
            else if (choice == 4)
```

```
        {
            string phone_no;
            cout << "Enter Phone No :-";
            cin >> phone_no;
            ptr->set_phone_no(phone_no);
        }
        else if (choice == 5)
        {
            string date_of_birth;
            cout << "Enter Date Of Birth :-";
            cin >> date_of_birth;
            ptr->set_date_of_birth(date_of_birth);
        }
        else if (choice == 6)
        {
            string designation;
            cout << "Enter designation :-";
            cin >> designation;
            ptr->set_designation(designation);
        }

        else if (choice == 7)
        {
            string str = ptr->get_username();
            cout << "Username -" << str << endl;
        }
        else if (choice == 8)
        {
            string str = ptr->get_name();
            cout << "Name -" << str << endl;
        }
        else if (choice == 9)
        {
            string str = ptr->get_email();
            cout << "Email -" << str << endl;
        }
        else if (choice == 10)
        {
            string str = ptr->get_phone_no();
            cout << "Phone number -" << str << endl;
        }
        else if (choice == 11)
        {
            string str = ptr->get_date_of_birth();
            cout << "Date of birth -" << str << endl;
        }
        else if (choice == 12)
        {
            string str = ptr->get_designation();
            cout << "Designation -" << str << endl;
        }
```

```
        else if (choice == 13)
        {
            int x = ptr->get_staff_id();
            cout << "Staff id -" << x << endl;
        }

        else if (choice == 14)
        {
            string str;
            int x;
            cout << "Enter the date " << endl;
            cin >> str;
            cout << "Enter <1,0> ( 1 for present and 0 for absent ) " << endl;
            cin >> x;
            ptr->add_date_for_attandence(str, x);
        }
        else if (choice == 15)
        {
            ptr->mark_todays_attandence();
        }
        else if (choice == 16)
        {
            string str;
            cout << "Enter Address " << endl;
            cin >> str;
            ptr->set_address(str);
        }
        else if (choice == 17)
        {
            string str = ptr->get_address();
            cout << str << endl;
        }
        else if (choice == 18)
        {
            int x;
            cout << "Enter salary" << endl;
            cin >> x;
            ptr->set_salary(x);
        }
        else if (choice == 19)
        {
            int x = ptr->get_salary();
            cout << "salary  " << x << endl;
        }
        else if (choice == 20)
        {
            vector<pair<string, int>> v;
            v = ptr->get_attendance_list();

            cout << "Date"
                 << "        "
```

```
                << "Status" << endl;
            for (auto x : v)
            {
                cout << x.first << "  " << x.second << endl;
            }
        }
        else if (choice == 21)
        {
            int x;
            string str;
            cout << "Enter the subject ";
            cin >> str;

            cout << "Enter the batch ";
            cin >> x;
            ptr->add_subject(x, str);
        }

        else if (choice == 22)
        {
            cout << "Subject list is as follows " << endl;
            ptr->get_subject();
        }

        else if (choice == 23)
        {
            ptr->conduct_exam();
        }
        else if (choice == 24)
        {
            int x;
            cout << "Enter the branch ";
            cin >> x;
            ptr->set_batch_no(x);
        }
        else if (choice == 25)
        {
            int x = ptr->get_batch_no();
            cout << "Branch " << x << endl;
        }

        else if (choice == 26)
        {
            int x = ptr->get_payment_done();
            cout << x << endl;
        }
        else if (choice == 27)
        {
            string book;
            cout << "Enter book name you want " << endl;
            cin >> book;
```

```
            string date;
            // input or use prabhav function to fetch todays date
            // now we are inputing
            cout << "Enter the todays date " << endl;
            cin >> date;
            obj_library.issue_teacher(book, date, id);
        }

        else if (choice == 28)
        {
            string book;
            cout << "Enter book name you want to return " << endl;
            cin >> book;
            string date;
            // input or use prabhav function to fetch todays date
            // now we are inputing
            cout << "Enter the todays date ( return date ) " << endl;
            cin >> date;
            obj_library.submit_teacher(id, book, date);
        }

        else if (choice == 29)
        {
            int id;
            cout << "Enter the roll no of the student you want " << endl;
            cin >> id;

            int check = 0;

            int size = student_list.size();

            student *pointer;

            auto it = student_list.begin();
            for (int i = 0; i < size; i++)
            {
                if (student_list[i].get_roll_no() == id)
                {
                    pointer = &student_list[i];
                    check = 1;
                    break;
                }
                it++;
            }

            if (check == 0)
            {
                cout << "The student with this id do not exist " << endl;
            }

            else
```

```
            {

                exam_marks m = ptr->give_marks();

                // pointer->exam.push_back(m); // direct cannot give the marks
                // so make a function in student

                pointer->add_exam_marks(m);
            }
        }

        else if (choice == 31)
        {
            ptr->display();
        }
        else if (choice == 32)
        {
            return;
        }

        else if (choice == 30)
        {

            int id;
            cout << "Enter your student id ";
            cin >> id;

            int check = 0;

            int size = student_list.size();

            student *pointer;

            auto it = student_list.begin();
            for (int i = 0; i < size; i++)
            {
                if (student_list[i].get_roll_no() == id)
                {
                    pointer = &student_list[i];
                    check = 1;
                    break;
                }
                it++;
            }

            if (check == 0)
            {
                cout << "The student with this id do not exist " << endl;
            }
            // check if the class teacher and student are of the same batch
            else if (pointer->get_standard() != ptr->get_batch_no())
```

```
                {
                    cout << "You are not the class teacher of this batch (student
belongs) --> so you cannot .... " << endl;
                }
                else if (pointer->get_standard() == ptr->get_batch_no())
                {

                    // can change

                    while (1)
                    {

                        cout << "MENU " << endl;
                        cout << "--------------------------------" << endl;
                        cout << "Press 1 for change username " << endl;
                        cout << "Press 2 for change name " << endl;
                        cout << "Press 3 for change email " << endl;
                        cout << "Press 4 for change phone_no " << endl;
                        cout << "Press 5 for change date_of_birth " << endl;
                        cout << "Press 6 for get username " << endl;
                        cout << "Press 7 for get name " << endl;
                        cout << "Press 8 for get email " << endl;
                        cout << "Press 9 for get phone_no " << endl;
                        cout << "Press 10 for get date_of_birth " << endl;
                        cout << "Press 11 for get standard " << endl;
                        cout << "Press 12 for get roll no " << endl;
                        cout << "Press 13 to get the percentage " << endl;
                        cout << "Press 14 to get the attendance list " << endl;
                        cout << "Press 15 to get the report-card " << endl;
                        int choice1;
                        cout << "Enter your choice " << endl;
                        cin >> choice1;

                        if (choice1 == 1)
                        {
                            string username;
                            cout << "Enter Username :- ";
                            cin >> username;
                            pointer->set_username(username);
                        }
                        else if (choice1 == 2)
                        {
                            string str;
                            cout << "Enter Student name :- ";
                            cin >> str;
                            pointer->set_name(str);
                        }
                        else if (choice1 == 3)
                        {
                            string email;
                            cout << "Enter Email :-";
```

```
            cin >> email;
            pointer->set_email(email);
        }
        else if (choice1 == 4)
        {
            string phone_no;
            cout << "Enter Phone No :-";
            cin >> phone_no;
            pointer->set_phone_no(phone_no);
        }
        else if (choice1 == 5)
        {
            string date_of_birth;
            cout << "Enter Date Of Birth :-";
            cin >> date_of_birth;
            pointer->set_date_of_birth(date_of_birth);
        }
        else if (choice1 == 6)
        {
            string str = pointer->get_username();
            cout << "Username -" << str << endl;
        }
        else if (choice1 == 7)
        {
            string str = pointer->get_name();
            cout << "Name -" << str << endl;
        }
        else if (choice1 == 8)
        {
            string str = pointer->get_email();
            cout << "Email -" << str << endl;
        }
        else if (choice1 == 9)
        {
            string str = pointer->get_phone_no();
            cout << "Phone number -" << str << endl;
        }
        else if (choice1 == 10)
        {
            string str = pointer->get_date_of_birth();
            cout << "Date of birth -" << str << endl;
        }
        else if (choice1 == 11)
        {
            int str = pointer->get_standard();
            cout << "Standard -" << str << endl;
        }
        else if (choice1 == 12)
        {
            int str = pointer->get_roll_no();
            cout << "Roll no -" << str << endl;
```

```
                    }
                    else if (choice1 == 13)
                    {
                        float x = pointer->get_percentage();
                        cout << "Percentage - " << x << endl;
                    }
                    else if (choice1 == 14)
                    {
                        vector<pair<string, int>> v;
                        v = pointer->get_attendance_list();

                        cout << "Date"
                             << "        "
                             << "Status" << endl;
                        for (auto x : v)
                        {
                            cout << x.first << "  " << x.second << endl;
                        }
                    }
                    else if (choice1 == 15)
                    {
                        pointer->show_report();
                    }
                    cout << "Do you want to perform more functions < y/Y for y
es > < n/N for no >";
                    char ch;
                    cin >> ch;

                    if (ch == 'y' || ch == 'Y')
                    {
                        // teacher_functions();
                        continue;
                    }
                    else
                    {
                        // return;
                        break;
                    }

                } // while loop
            }
        }

        cout << "Do you want to perform more functions < y/Y for yes > < n/N f
or no >";
        char ch;
        cin >> ch;

        if (ch == 'y' || ch == 'Y')
        {
            // teacher_functions();
```

```
                continue;
            }
            else
            {
                // return;
                break;
            }

        } // while loop

    } // else condition

} // complete function
```

All the functions of teacher can be used here also

With the additional features

Choice 24 → change the branch of which you are the class teacher

Choice 25 → get the branch of which you are the class teacher

Choice 29 → give marks to student

The marks of the student of a particular batch can be added only by that batch class teacher

(Real life assumption → we see in real life that all the different subject teachers give the marks to the class teacher who then adds the marks to the marks to the report card of respective students.)

Working

The class teacher is using his/her functions right now, so he/she will be asked to input the id of the student who you want to change the marks

The class teacher enters the id

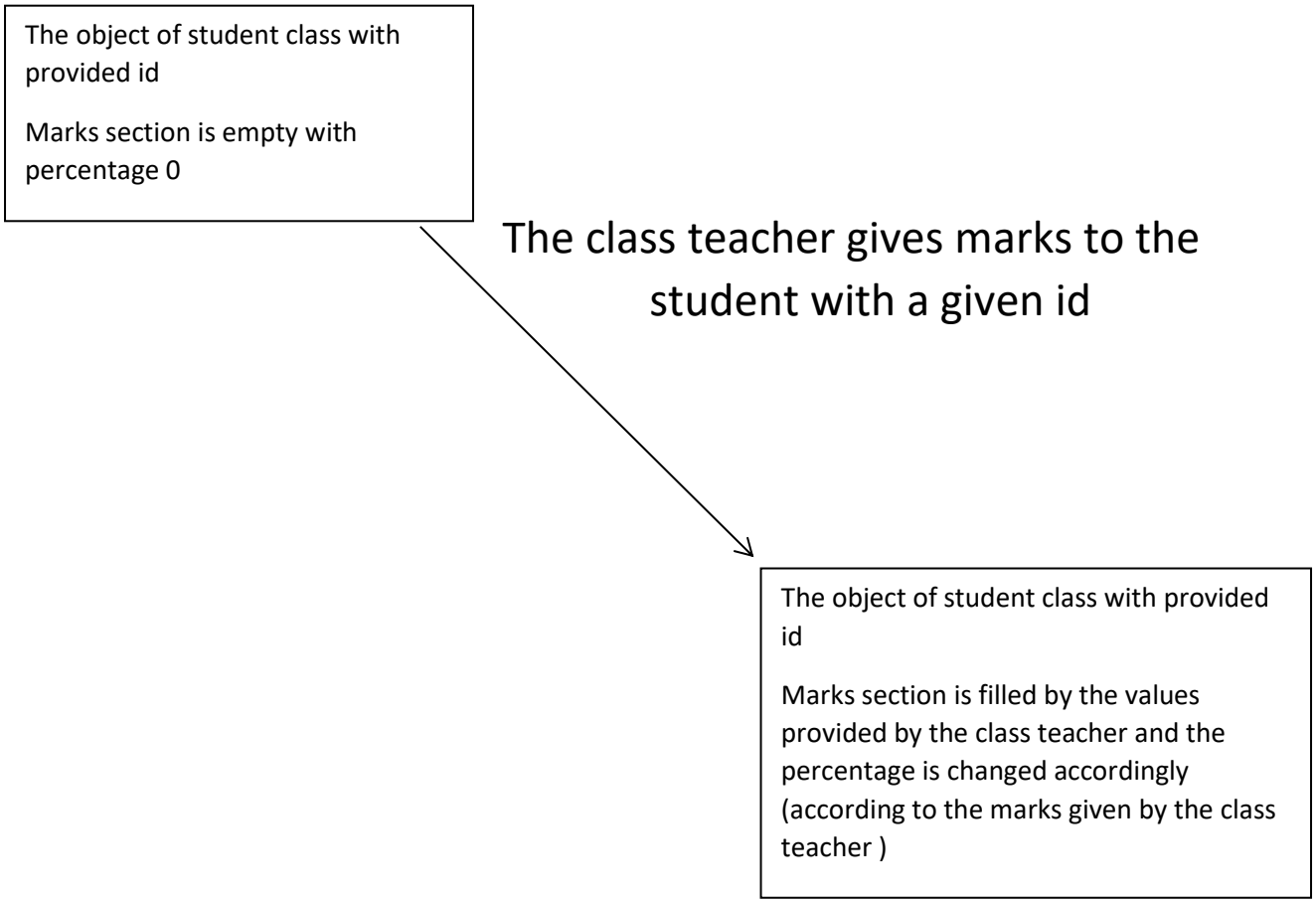
The our system finds the student with this id

Case 1 → if the student is not found → student not found displayed

Case 2 → if the student is found

Case 2.1 → if student standard and the class teacher batch (standard) do not match → you are not the class teacher of the class in which the student with provided id is present so you cannot give him marks

Case 2.2 → if id matches and the standard matches then the class teacher can give the marks of the student and the following state transition of the object takes place



Note → Now you can log out from CLASS TEACHER access and enter the system again with STUDENT access mode and with given id and check whether the marks have been added and the percentage have been changed or not → you will find that the percentage and marks have both been updated .

Choice 30 → change entities of student

The class teacher a special function to change the fields of a student of his/ her class

The working is similar to the Choice 29 , in which we were giving the student marks now we are changing fields of the student

Security Features

The main feature of OOP is data hiding, so by using OOP approach we ensure the security of our system.

Other than that we have made a complete login system which ensures 2 things

1. A person not knowing one of the seven username and password cannot enter the system
2. The 7 users which we provide the functionality cannot use each other functionality

Advantages

1. We have provided functionality to all the users which you will find linked with the school
2. Our system is secure
3. For all the 7 users we have tried to cover all major functionality
4. We have kept our system close to real world so it can be used effectively by all the users
5. Apart from students and teachers we have two more important management systems working inside our project
   C. Bank management system → managing the school amount details
   D. Library management system → managing the school library

Disadvantages

1. The system lacks permanent storage so as soon as you close the system all the data you saved is lost. We understand that this is important feature but due to lack of time we failed to implement them. But we have focussed on the design part and implementation lacking this feature of permanent storage which can be achieved using Filing System (storing data in .txt files) or Database management system (DBMS).
2. We haven’t implemented the GUI version of the project so our projects runs in terminal only.

Future scope and improvements required

As discussed in the disadvantages if we manage to provide permanent storage to our system and convert it to GUI version then our system fulfils all the requirements of the Complete School Managing System which can be used to actually manage the major functionalities that take place in school.

In our project there is a scope of using multithreading that is running two different part of the code at the same time.

For example

1. The bank and the library can work at the same time
2. The student and the teacher can work at the same time and so on

THANK YOU