**PROJECT REPORT**

**ON**

**SCHOOL MANAGEMENT SYSTEM**



**SUBMITTED BY:-**

**Under the guidance of Academic Head,Submitted in partial fulfillment of the requirement for qualifying NIELIT ‘A’ Level**

**COVERING LETTER**

**Managing Director,**

**NIELIT,**

**6,CGO Complex**

# Lodhi Road

# New Delhi-110003

Sir,

I am submitting my ‘A’ Level Project for evaluation. Details of my Registration and postal address, etc are as under:

**Regn. No: - Level: - ‘A’**

**Name : -**

Email Address : -

Mobile :-

**CERTIFICATE**

This is to certified that the work embedded in the project report entitled “SCHOOL MANAGEMENT SYSTEM” is own account of work carried by for **NIELIT ‘A’** **LEVEL** student as a Direct Candidate has been carried out under my direct supervision and guidance and to the best of my knowledge the work has not been submitted else where for the owned of any other degree.

His efforts on this project have been found satisfactory for obtaining the “NIELIT ‘A’ LEVEL CERTIFICATE”.

**Authorized Signature**

**UTTARAKHAND**

**Submitted by: Authorized Signature**

**ACKNOWLEDGEMENT**

I want to express sincere thanks and gratitude to my Supervisor, for helping me in making of this project a true value added experience, which shall help me immeasurably in my career development.

Her knowledge and teaching broaden my mind and made me to think on all aspect. I can not imagine myself without her help and valuable guidance in the development of this project. I am extremely obliged for her patience and constant encouragement throughout the project and uplifting my spirits, I thanks her for all support and encouragement provided by her and consistent guidance and keen interest in the whole work led to the fulfilling the objective of this assignment. It also enabled me to acquire rich and valuable experience of developing real world application.

I also wish to express my gratitude to NIELIT who gave us this opportunity to make such projects which are relevant for present world and also which enhance our minds. I also like to express my gratitude to my family and friends for their support an encouragement throughout the project.

**TABLE OF CONTENT**

1. Introduction
2. Objective
3. Problem Definition
4. System Analysis & User Requirement
5. Software Development Life Cycle
   1. System Planning(Pert Chart)
   2. Feasibility Study
   3. Hardware & Software Requirement
6. Cost & Benefit Analysis
7. Project Designing
   1. Data Flow Diagram
   2. Data Dictionary
   3. ERD Diagram
   4. Input & Output Screens
8. Source Code
9. System Implementation
10. Testing
11. System Maintenance
12. Computer Security
13. Future Scope of the Project
14. User Manual
15. Conclusion
16. Bibliography

**INTR****ODUCTION**



SCHOOL MANAGEMENT SYSTEM is software that provides a computerized system. it is developed for converting of a manual system to fully computerized system with its help we can store record in the files directly we need not to do any thing manually we can even extract record directly from these files. It also help us by removing all those mistakes which occur while entering data since, if we enter wrong id it wont allow us to enter the data into the files, now their would no problem in finding the record from books we can simple type their id and record of that id would be generated.

The program would take care of every thing the operator just have to insert the information rest of the work would be done by the program itself i.e. Generating id automatically, Adding of student , Transfer certificate given to the student .It also can be used for entering student marks and saving it in the files for future use. This software also can be used while receiving fee from the student, fine and giving salary to the staff member and many other things would be computerized. A user can generate any information by selecting that choice from the menu it makes our work easy and more accurate than what use to be done manually.

This project is step forward towards the enhancement of the scope of the computer .I am trying to make software for the school management which would make the school management easy to handle and computer are being used to solve all the aspect of the school management.

**OBJE****CTIVE**

My main objective is to provide computerized School Management System and report generation subsystem, which can provide the facility to register student provide them transfer certificate, storing of marks of student class wise and also Staff information consist of both of teachers and other staff members, by help of this software we can also store fees and salary information received or provided by the institute. It is also capable of showing and storing fine information into the separate fine file. We can find out what amount of total fees, fine, salary paid for a particular month easily and efficiently without any problem we need not to search a record of a staff member or a student from a pile of files we just need to enter his or her id and can find about all the information for a particular period any time, other essential feature of this software is its security feature which is at both user level and system level. System level security is that without knowing password and user id a user can not enter into the software and on the other hand if user level if by mistake user enter wrong id the software wont take it and also the auto generate id facility which gives full proof system i.e. already given id wont be allotted to any other person, in whole the system provides us with most efficient and effective working environment.

The system objective to provide easy to use CUI based system, which does not require a heavy OS to be installed; the software prints details on the screen.

This software would make the work easily and simple so that any one could get any information or can do his work faster, accurately and systematically way electronically rather doing all the work manually.

**PROBLEM** **STATEMENT**

‘SCHOOL MANAGEMENT SYSTEM’ software providing the general computerized System. The information about the student, staff, salary, fee and fine of the school is maintained properly and every student and staff member have a specific id by which we can find all the information about them and also their should be password and id so that only authorized person access the software.

It should be able to handle salary of the employees and student fee report. It should also be able to store student marks and also information about the staff and teachers. It should also be able to handle fine received by the studentand also the complete information about the sport.

There is growing realization that the more widespread use of information technology to increase efficiency and enable changes in school management processes could do much to improve the performance of the school management system, within the bounds of appropriate measures to protect the confidentiality of private information. Hence the higher authorities of school management have decided to go for a computer based system.

**Requirement Summary**

Depending on the decision given by the management on the Computer Based School Management System is to be developed.

**Following are the requirements based on which the system needs to be developed.**

1. Admission of the student and allotting id to them.

2. Providing TC to the student.

3. Storing Student marks and showing result both class and student wise.

4. Security of the software by putting password and id in it.

5. Account section storing information about the fee, fine and salary.

6. Information about the Staff and Teachers separately and providing separate unique id to them.

7. Sport section containing information of student participated and the awards or prizes won by them at what level.

8. And also information about teacher who are in charge of sport.

9. History of the school should be present in the software so that new comer can know about its foundation information.

**SYSTEM ANALYSIS AND DESIGN**

**TRANSFORMATION**

DESIGN

**SRS DESIGN** **SPECIFICATION**

TECHNICAL

DESIGN

CONCEPTUAL

DESIGN

**CUSTOMER SYSTEM BUILDER**

S/W DESIGNING is a highly creative process of s/w development cycle where the designer plan how a s/w system should be produce to make a functional reliable and reasonably easy to understand, SRS tell us what a system does and became the input for the design.

It is a process which help us how a system works. SRS justified with the designed specifications. The process of s/w design involves the transformation of idea in to detail, implementation, and description with the objective of satisfying the s/w requirement in to working system the designer must satisfy both customer as well as system builders.

A good s/w design needs to fulfill :-

* Correct and complete
* Understandable
* At the right level
* Maintainable
* Flexibility

**These activities are shown in figure and discussed further:**

Government rules & Regulation

Organization

Consumers

Top Management

Union

User

Competition

System Analysis

Recognition of Need

Feasibility Study

Design

Analysis

Post Implementation

Implementation

Maintenance

**The Linear Sequential Model**

The Linear Sequential Model is the oldest and the most widely used paradigm for software engineering. This Model is sometimes called “the Classic life Cycle” or “the Waterfall” Model, which suggests a systematic, sequential approach to software development that begins at the system level and progresses through analysis, design, coding, testing & support.

Diagrammatic representation of the Linear Sequential Model is as:

Analysis

Design

Code

Implementation

The Linear Sequential Model

**INITIAL INVESTIGATION:-**

The objective of determining the validity of the user request for the candidate system and whether a feasibility study should be conducted. The objective should be understood within the framework of the organization MIS(Management Information System).

**The basic steps that were followed during the Initial Investigation:-**

**Recognition of the Need:**

To recognize what the user need from the system. The success of the system depends largely on how accurately a problem is defined, through investigation , and properly carried out through the choice of the solution .User need identification and analysis are concerned with the user needs rather than what he\she wants.

**What is the problem?:**

What problem the users are facing which has motivated them to ask for the candidate system? What all short coming do the existing system is having? What all changes does the user request?

**INFORMATION GATHERING**

Review Literature

Procedures and Forms

On-Site

Observation

Interviews

Questionnaires

**On-Site observation Technique**

The major objectives of on-site observation are to get as close as possible to the ‘real’ system being studied. The following questions were answered during the on-site observation technique:-

**\*** What kind of system is it?

\* What does it do?

\* Who runs the system?

\* Who are the important people in it?

\* What is the history of the system?

\* How did it get to its present stage?

\* What kind of system is it in comparison with other systems in the Organization?

**Information about the User Staff**

In summary, the major focus is to find out what people the analyst is going to be dealing with and what each person expects to get out of a candidate system before it goes through design and final implementation.

The application using staff is the well-qualified employees. They are very good in their respective fields. But less about the use of the application, so the system has to be developed so that it can help them to understand the application better and faster.

**QUESTIONAIRE**

**Problems Faced:-**

Before designing any project we should know why we are making it, as we all know that “NECESSITY IS THE MOTHER OF EVERY INVENTION”. As far as this project is concerned, it deals with the problems related to the School Management System of an institute.

**The problems that can arise are as follows:**

**Problem 1:** The first problem that arises is that the maintaining the electronic database is very much Time consuming as all the work is done manually.

**Problem 2:** The maintenance of the records all student & other staff members of the institute are very difficult.

**Problem 3:** There is a difficulty in maintaining the records of all the faculties i.e. (Teachers).

**Problem 4:** The maintenance of the records of salary, fee and fine paid or received by the institute is quite difficult.

**Problem 5:** The maintenance of the records of the position of the student according to the marks and past result is very difficult to maintain and present when demand.

**Problem 6:** The problem that may arise about staff information.

**Problem 7:** The sport session information storing and retrieving when demanded**.**

**Solution to the problems**

Once the analysis of problem is completed, we have a firm understanding of what has to be done. In this part of the project, we will produce the details that describe how the database will meet the solutions to the problems stated or requirements identified during the analysis process. This stage is often referred to as the LOGICAL DESIGN designing.

**USER REQUIREMENTS:**

Besides all the jobs that were being done in the existing system, the user expected certain other features to be added to the package. Since the existing system is manual, reports are not generated but in the proposed computerized system the reports are required at different queries.

In the manual system, auto generated ids is not present which is present in this software by which any person information can be generated at any time easily and efficiently**.**

**PERFORMANCE REQUIREMENTS:**

**\* The following performance characteristics were taken care of while developing the system*:***

**User Friendliness**

The system is easy to learn and understand. A native user can also use the system effectively, without any difficulty.

**User Satisfaction**

The system is such that it stands up to the user’s expectations.

**Response Time**

The response time of all the operations is good .This has been made possible by careful programming.

**Error Handling**

Response to user errors and undesired situations have been taken care of ensure that the system operates without halting**.**

**Safety and Robustness**

The system is able to avoid or tackle disastrous action. In other words, it should be foul proof. The system safety guards against undesired events without human intervention.

**SOFTWARE DEVELOPMENT LIFE CYCLE**

A software life cycle is a series of identifiable stages that a software product undergoes during its life time. This project “E-REPOSITORY SYSTEM” usually starts with a feasibility study stage, and then requirement analysis and specification, design, coding, testing and maintenance. In this model the life cycle model is ‘Iterative Waterfall Model’

**ITERATIVE WATERFALL MODEL**

**The waterfall model divides the life cycle of a software development process into phases as shown:-**

**FEASIBILITY ST**

**AND SPECIFICATION**

**TESTING**

The classical waterfall model is an idealistic one since it assumes that no development error is ever committed by the engineers during any of the life cycle phases .However in practical development environment, the engineers do commit a large numbers of errors in almost every phase of life cycle. Resource of the many can be: oversight, wrong assumptions, use of inappropriate technologies, communication gape among the project engineers etc. These defects usually get detected much later in the life cycle. For example a design defect might go unnoticed till we reach the coding and trusting phase. Once, a defect is detected, the engineers need to go back to the phase where the defect had occurred and redo some of the work done during that phase and subsequent phase to correct the defect and its effect on the later phases.

Therefore in any practical software development work it is not possible to strictly follows the classical waterfall model. Feedback paths are needed in the figure, to allow for the correction of the errors committed during the phase that are detected in later phase.

Though errors are inevitable in almost every phase of development it is desirable to detect these errors in the same phase in which they occur. However, it may not always be possible to detect all errors in the same phase in which they occur. Nevertheless ,the error should be detected in the design phase itself this problem can be taken care of easily than if the error is identified, say, at the end of the integration and system testing phase. In the later case, it would be necessary not only to rework, but also to appropriately redo the coding and the system testing, thus incurring higher cost. The principle of detecting errors as close to their points of introduction as possible is known as the phase containment of errors. This is an important software engineering principle.

**SYSTEM PLANNING**

**PERT Chart**

A PERT chart presents a graphic illustration of a project as a network diagram consisting of numbered nodes (either circles or rectangles) representing events, or milestones in the project linked by labelled vectors (directional lines) representing tasks in the project. The direction of the arrows on the lines indicates the sequence of tasks. In the diagram, for example, the tasks between nodes 1, 2, 4, 8, and 10 must be completed in sequence. These are called dependent or serial tasks. The tasks between nodes 1 and 2, and nodes 1 and 3 are not dependent on the completion of one to start the other and can be undertaken simultaneously. These tasks are called parallel or concurrent tasks. Tasks that must be completed in sequence but that don't require resources or completion time are considered to have event dependency. These are represented by dotted lines with arrows and are called dummy activities. For example, the dashed arrow linking nodes 6 and 9 indicates that the system files must be converted before the user test can take place, but that the resources and time required to prepare for the user test (writing the user manual and user training) are on another path. Numbers on the opposite sides of the vectors indicate the time allotted for the task.

Specification

Part 10

Coding Part 25

Design Structure

Part 10

Write User

Manual 7

Code GUI

Part 10

Integration

Part 10

Design GUI part 5

Finish

0

**FE****ASIBILITY STUDY**

The Manual SCHOOL MANAGEMENT SYSTEMS are complicated and hectic, as everything has to be maintained on the paper. We know that human operated systems are erroneous systems, and the School Management System available in the market is so costly and it requires the special crash courses to undergo in order to operate them. This Software is best in the sense that it is a low cost system, as it could be run on any existing personal computer running DOS/Windows9x/Xp/2000 operating system. Further to run this system one does not need to install any specific OS/Software/DBMS system to support the system.

**Economical Feasibility:**

I found my software to fit in the existing low cost Personal Computer, which is having at least DOS or windows98/Xp/2000 as its operating system. System implementation do not requires any specific software/hardware to be installed. So economically it is cost effective software which it self has a low cost and do not require a user to spent more.

**Technical Feasibility:**

This system do not require any technical training to be provided to the users and no new technology has to be purchased, the system is fit with the existing working PC running DOS/Windows98/Xp etc.

**Behavioral Feasibility:**

A system is good if it is user friendly. The SCHOOL MANAGEMENT SYSTEM software is presenting itself in a Character User Interface, which is supported by plain text menu operated environment to give an easy to use end user experience.

**SCHOOL MANAGEMENT SYSTEM made for operating with keyboards hence its feasible by its simple behavior.**

**Hardware & Software Requirement**

**There are several things to do before selection:**

1. Define system capabilities that make sense.

2. Specify the magnitude of the problem.

3. Assess the competence of the in-house staff.

4. Consider hardware and software as a package.

5. Develop a time frame for selection.

6. Provide user indoctrinations.

**The criteria for software selection are:**

1. Reliability : Gives consistent results.

2. Functionality : Functions to standards.

3. Capacity : Satisfies volume requirements.

4. Flexibility : Adapts to changing needs.

5. Usability : Is user friendly?

6. Security : Maintains integrity and prevents

unauthorized access.

7. Performance : Delivers as expected.

8. Serviceability : Has good documentation and vendor

support.

9. Ownership : Has right to modify and share use of

package.

10. Minimal Costs : Is affordable for intended Application

**OPERATING ENVIRONMENT**

Minimum Hardware Requirements

* Microprocessor : Pentium-3 166MHz
* RAM : 256 MB

* Hard Disk : 2.1 GB
* Monitor : SVGA Monitor

**Software Requirements**

* Operating System : Windows 7,8,10
* Front-End-Tool : Turbo c++
* Back End Tool : C File Handling

**SOFTWARE** **PROFILE**

**Front-En****d (C++ language)**

The front-end used to develop this system is Turbo C++.

**About C++ Language**

C++ programming language was developed at AT &T Bell laboratories by Bajarne Stroustrup in 1979.C++ begin as the expanded version of C.C++ has undergone three major revisions with each revision improving the language. the reason behind the success of c++ is that it support object oriented technology Object oriented technology is regarded as ultimate paradigm for the modeling of information ,be that the data or logic.

C is programming language developed at Bell Laboratories of USA in 1972. It was designed and written by Dennis Ritchie. In the late 70’s C begin to replace more familiar language of that time like PL/I, ALGOL, etc. C seems so popular is because it is reliable, simple and easy to use. Moreover, in an industry, where newer languages, tools and technologies merge and vanish day in and day out, a language that has survived for more than 3 decades has to be really good.

C Language is the most popular tool for programming creating fast business solutions for Windows. C helps developers quickly create and deploy applications and program easily for the C language programming tools and techniques.

Procedure Oriented Programming is popularly called as POP. On one hand, POP is a programming paradigm its own right and on the other, it is a set of software engineering tools, which used to build more reliable and reusable systems.

**DIFFRENCE BETWEEN POP &OOPS**

In a pop language a program is a list of a instruction where each statement tells the computer to do something .this language provide facilities for passing arguments to functions(sub program)and returning values from functions. Procedure programming does not model real world problem very well.

OOPs approach views a problem in terms of objects involve rather than procedure for doing it .OOPs took the best ideas of structured programming and combined them with different and new concepts a program in general can be organized in one or two ways 1st around its code 2nd around its data. Using procedure is organized around its code, OOPsare organized around its data.

In Oops we define a data type by creating a class and when the class is defined we produce its instance

**ADVANCE FEATURES IN C++**

It has been developed with a view to overcome the drawbacks of conventional programming approaches. These general concepts of oops are:-

1. Data Abstraction.
2. Data Encapsulation.
3. Inheritance.
4. Polymorphism.
5. Dynamic binding.
6. Object.
7. Class.

**COST & BENEFIT**

This software won’t be expensive because the software used for making up of this software is the turbo c++ which is neither expensive nor difficult to use. The employees who are going to use this software need not to take any kind of training. They can easily understand the software by just going through the user manual.

On the other hand we need not to use any secondary software like oracle or the Microsoft SQL for storing of the record since the information is stored into the files with the use of file handling. This software can be used by any scale institute whether large, middle or small institute.

This software provides us with many benefits such as removal of extra burden from employees who have to make separate files to store the information about the student, staff and other financial information and also for every year. Software remove all the problem faced while registration of student or recruitment of the staff member, it also take care of the financial department which include salary to staff members, fee or fine from the student the extra feature in it is that it store information about the sport and the staff member consisting of both the teachers and other staff members under sports heading we store the information about team members and the staff person who the look after that sport and even the history of the institute.

**PROJECT DESIGNING**

**DATA FLOW DIAGRAM (DFD)**

A DFD, also known as a “bubble chart” has the purpose of clarifying system requirements and identifying major transformations that will becomes programs in system design.

So it is the starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of detail. A DFD consist of a series of bubbles joined by data flows in the system. It is very useful to make user understand the working of the system.

**CONT****EXT LEVEL**

The context diagram is the highest level view of the system, it contains only one process, no data stores, and a number of data flows and also a number of source/sinks. The single process represents the entire system and its labeled “o”. The source/sinks represent its environmental boundaries. Since the data stores conceptually inside the one process, no data stores appear on a context diagram.

It is an overview of an organizational system that shows the system boundaries, external entities that interact with the system, and the major information flows between the entities and the system.

In the figure, the context diagram of the System is shown. It consists of the following entities and the following data flows.

**Symbol used in DFD:-**

|  |  |
| --- | --- |
| **SYMBOL** | **MEANING** |
|  | **External entity and Output** |
|  | **Process** |
|  | **Data Flow** |
|  | **Data Store** |

**CON****TEXT LEVEL DFD**

Users

input \_data

response

**LEVEL 1 DFD**

student \_id

student

registration

admission

saff

registration

student \_id

Staff\_info

sport

fine

marks

fee\_slip

sport\_info

sport report

pay\_slip

Salary\_info

**LEVEL 2 DFD**

fine\_slip

fine

marks

fee\_info

result

fee\_slip

fine\_request

fine\_response

enter\_marks

fee\_response

fee\_request

reponse

**ENTITY R****ELATOINSHIP DIAGRAM (ERD):-**

The entity –relationship (E-R) data model is based on a perception of a real world that consists of a collection of basic objects, called entities & of relationships among these objects. an entity is a “thing” or object in the real world that is distinguishable from other objects .for example ,each person is an entity and bank accounts can be considered as entities .

Entities are described in a database by a set of attributes.

Relationship is an association among several entities. The set of all entities of the same type and the set of relationships of the same type are termed as entity set and relationship set respectively.

The overall logical structure (schema) of a database can be expressed graphically by an E-R diagram, which is built up from following components:

1. Rectangles, which represents entity sets.
2. Ellipses, which represents attributes.
3. Diamonds, which represents relationships among entity sets.
4. Lines, which link attributes to entity sets and entity sets to relationships

**Header file used**

|  |  |
| --- | --- |
| Stdio.h | Standard Input Output |
| Conio.h | Console I/O functions |
| Stdlib.h | Standard Library |
| String.h | Standard String Library Functions |
| Ctype.h | Character Manipulation Functions |
| Fstream.h | File Stream |

**DATA DICTIONARY**

I have used structures as data structure to organize the data into file. The system reads the data from user in its form and uses file handling to make changes to the binary database files following are the structure used.

###### CLASS STUDENT

###### Structure # 1: stud

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Name** | **Data Type** | **Description** | **Length** |
| 1. name | Char | Name | 25 |
| 1. mother | Char | Mother name | 25 |
| 1. father | Char | Father name | 25 |
| 1. dob | Char | Date of birth | 10 |
| 1. add | Char | Address | 50 |
| 1. contact | Char | Phone no | 10 |
| 1. father\_occp | Char | Occupation | 25 |
| 1. domicile | Char | Domicile | 25 |
| 1. school\_name | Char | Old school name | 25 |
| 1. Tc | Char | Transfer certificate | 5 |
| 1. Religion | Char | Religion | 25 |
| 1. Disease | Char | Disease | 25 |
| 1. Blood\_group | Char | Blood group | 5 |
| 1. Id\_student | Int | Student identification | 2 |
| 1. Father\_income | Float | Father income | 4 |
| 1. Mother\_income | Float | Mother income | 4 |
| 1. Pert | Float | Percentage | 4 |

###### Structure # 2: trans

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Name** | **Data Type** | **Description** | **Length** |
| 1. name | Char | Name | 25 |
| 2. char\_certific | Char | Character certificate | 5 |
| 3. id | Int | Student identification | 2 |
| 4. class | Int | Class | 2 |
| 5. date | Char | Date of tc | 10 |

###### CLASS EXAM

###### Structure # 1: mark

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Name** | **Data Type** | **Description** | **Length** |
| 1. mark\_id | Int | Student Identification | 2 |
| 2. clas | Int | Class | 2 |
| 3. hindi | Float | Hindi marks | 4 |
| 4. eng | Float | English marks | 4 |
| 5. bio | Float | Biology marks | 4 |
| 6. chem | Float | Chemistry marks | 4 |
| 7. phy | Float | Physics marks | 4 |
| 8. com | Float | Commerce marks | 4 |
| 9. comp | Float | Computer marks | 4 |
| 10. eco | Float | Economic marks | 4 |
| 11. acc | Float | Account marks | 4 |
| 12. hist | Float | History marks | 4 |
| 13. geog | Float | Geography marks | 4 |
| 14. social | Float | Social marks | 4 |
| 15. math | Float | Mathematics marks | 4 |
| 16. art | Float | Art marks | 4 |
| 17. science | Float | Science marks | 4 |
| 18. total | Float | Total marks | 4 |

###### CLASS ACCOUNT

###### Structure # 1: fine

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Name** | **Data Type** | **Description** | **Length** |
| 1. fine\_id | Int | Student identification | 2 |
| 2. type | Char | Type of fine | 25 |
| 3. fine | Float | Fine | 4 |

###### Structure # 2: salary

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Name** | **Data Type** | **Description** | **Length** |
| 1. sal\_id | Int | Staff/teach identification | 2 |
| 2. basic | Float | Basic pay | 4 |
| 3. da | Float | Da | 4 |
| 4. hra | Float | House rent allowance | 4 |
| 5. pa | Float | Provident | 4 |
| 6. mc | Float | Medical allowance | 4 |
| 7.teach | Char | Teacher/staff | 25 |

###### Structure # 3: fee

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Name** | **Data Type** | **Description** | **Length** |
| 1. fee\_id | Int | Student identification | 2 |
| 2. misc | Float | Miscellaneous fee | 4 |
| 3. tuition | Float | Tuition fee | 4 |
| 4. bus | Float | Bus fee | 4 |
| 5. magazine | Float | Magazine fee | 4 |
| 6. tour | Float | Tour fee | 4 |
| 7. sport | Float | Sport fee | 4 |

###### CLASS MISCELL

###### Structure # 1: fine

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Name** | **Data Type** | **Description** | **Length** |
| 1. sport\_id | Int | Sport identification | 2 |
| 2. game | Char | Game name | 25 |
| 3. head | Char | Head coach name | 25 |
| 4. sub\_head | Char | Sub coach name | 25 |
| 5. captain | Char | Captain name | 25 |
| 6. vice | Char | Vice captain | 25 |
| 7. other | Char | Team members name | 50 |
| 8. level | Char | State/University | 25 |
| 9. award | Char | Award | 25 |
| 10. position | Char | Position | 25 |

###### Structure # 2: staff

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Name** | **Data Type** | **Description** | **Length** |
| 1. staff\_id | Int | Staff identification | 2 |
| 2. name | Char | Staff name | 25 |
| 3. add | Char | Address | 4 |
| 4. phone | Char | Phone no | 10 |
| 5. retired | Char | Retired | 5 |
| 6. job | Char | Job | 25 |
| 7. post | Char | Post | 25 |

###### Structure # 3: teacher

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Name** | **Data Type** | **Description** | **Length** |
| 1. teach\_id | Int | Teacher identification | 2 |
| 2. name | Char | Staff name | 25 |
| 3. add | Char | Address | 4 |
| 4. phone | Char | Phone no | 10 |
| 5. qualification | Char | Qualification | 25 |
| 6. subject | Char | Subject | 25 |
| 7. taught | Char | Subject taught | 25 |
| 8. clas\_teacher | Char | Class teacher | 5 |
| 9. activity | Char | Activity incharge | 25 |

**E R DIAGRAM**

**employee**

**Salary**

**Mark**

**staff**

**Staff**

**Sports**

**Coach**

**Stud**

**Players**

**Trans**

**TC**

**Teacher**

**Fee’s**

**Fine**

**Result**

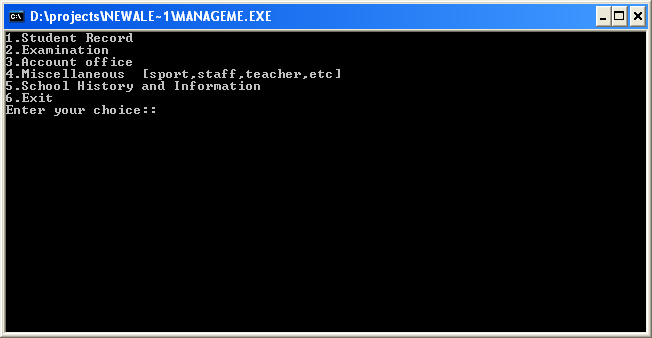
**Fee**

**Fine**

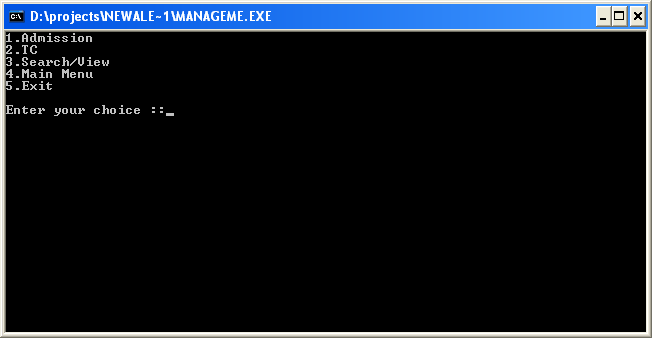
**INPUT & OUTPUT SCREENS**

**Main page**

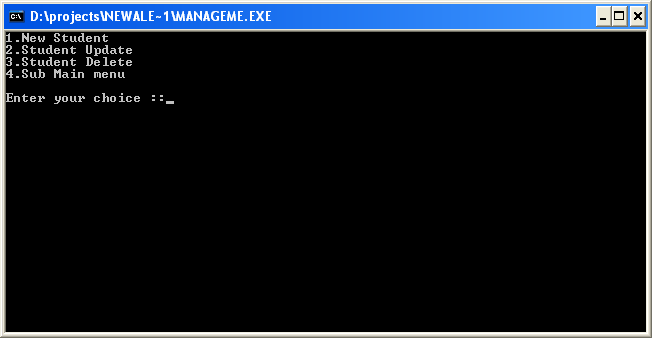
**Main menu**

****

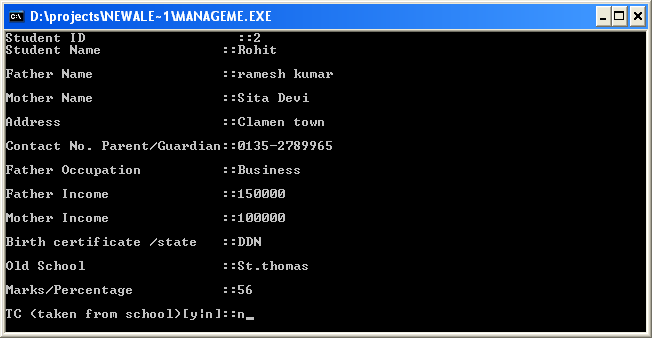
**Student Main Menu**

****

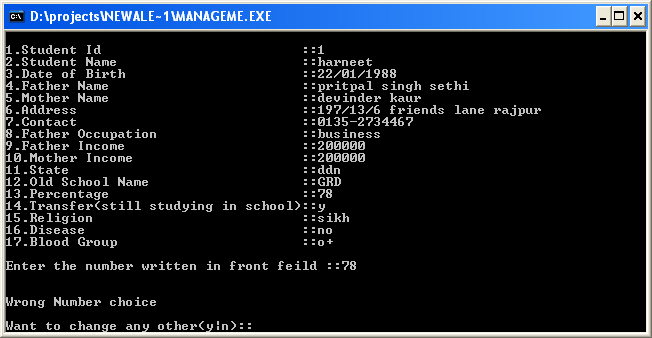
**Admission Main Menu**

****

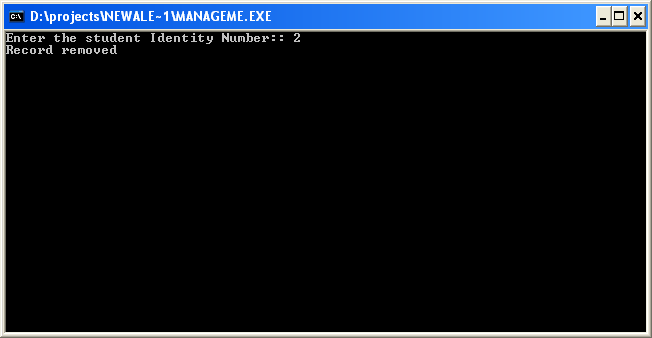
**Registration of Student**

****

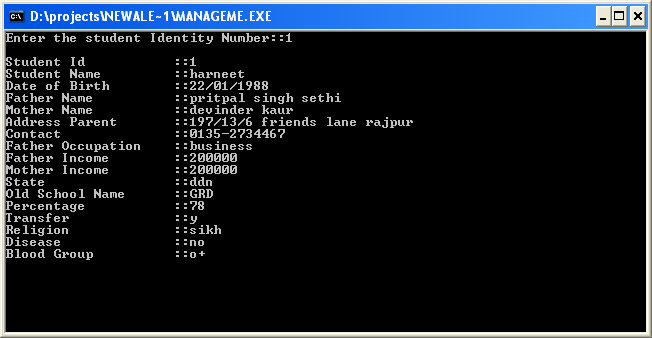
**Update of Student**

****

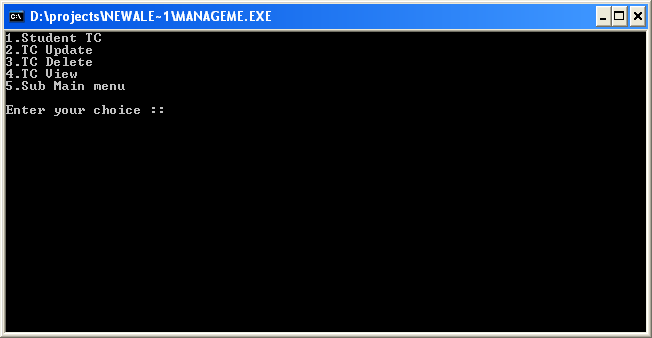
**Delete Student Record**

****

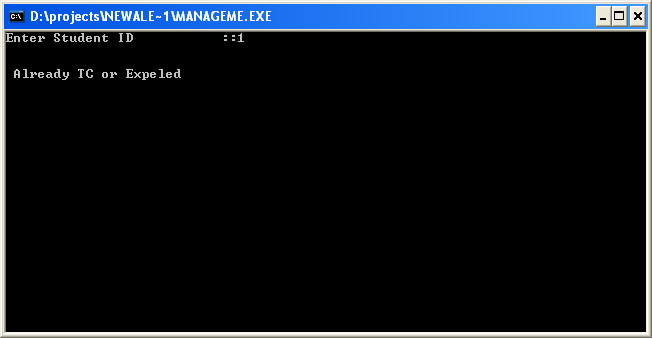
**View Student Record**

****

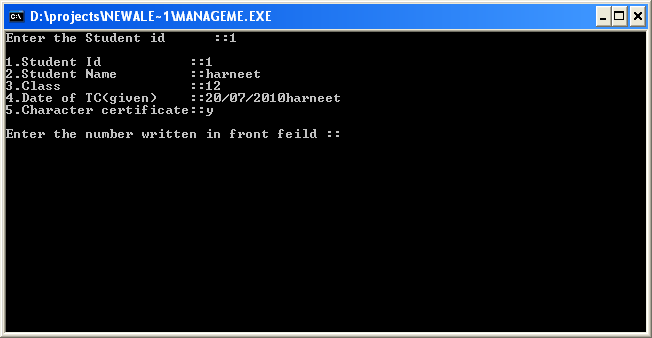
**Tc Main Menu**

****

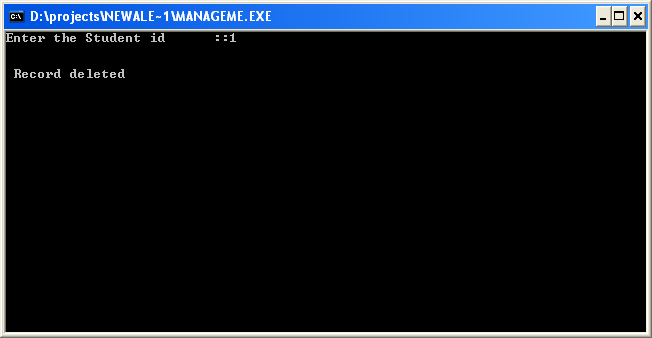
**Tc to Student**

****

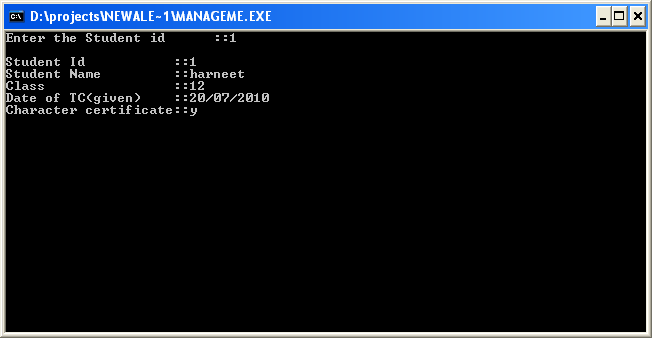
**Update of TC**

****

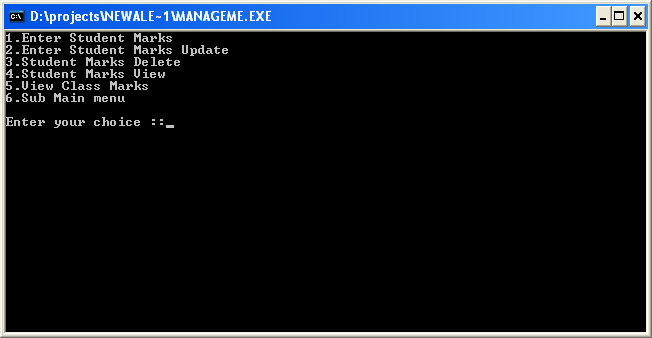
**Deletion of Tc**

****

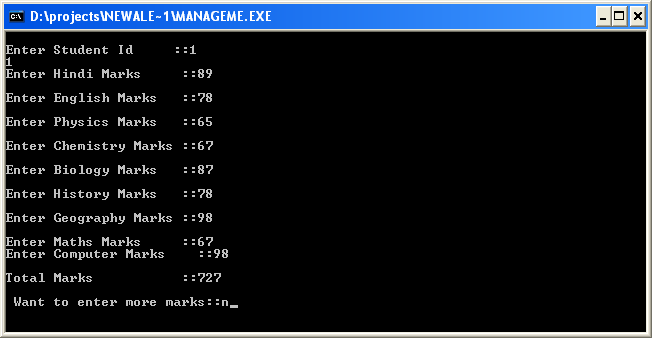
**View Tc Detail**

****

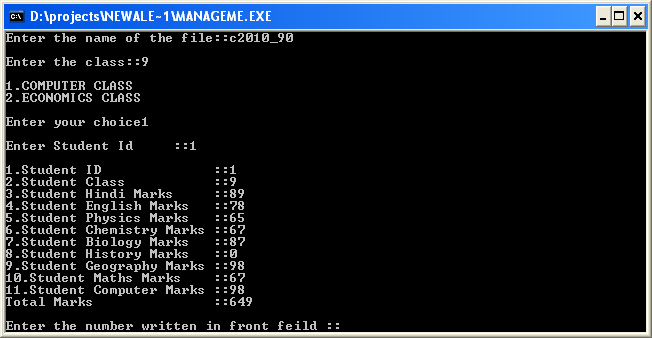
**Examination Main Menu**

****

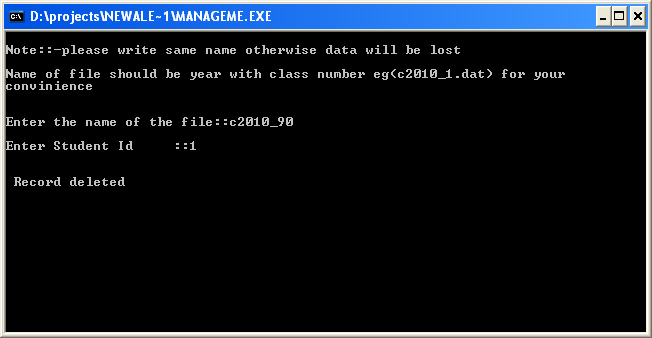
**Enter Student Marks**

****

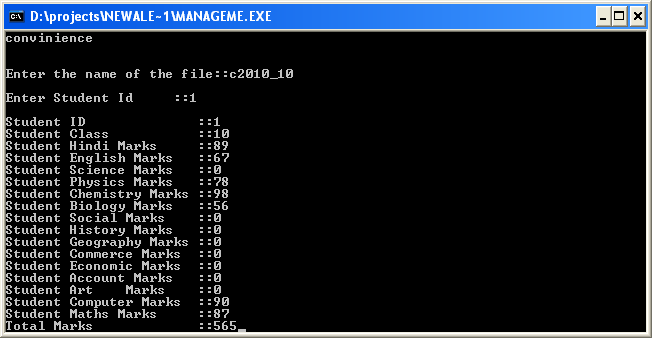
**Update Student Marks**

****

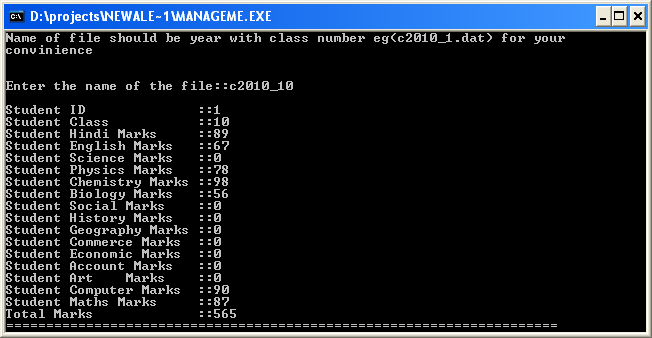
**Delete Student Marks record**

****

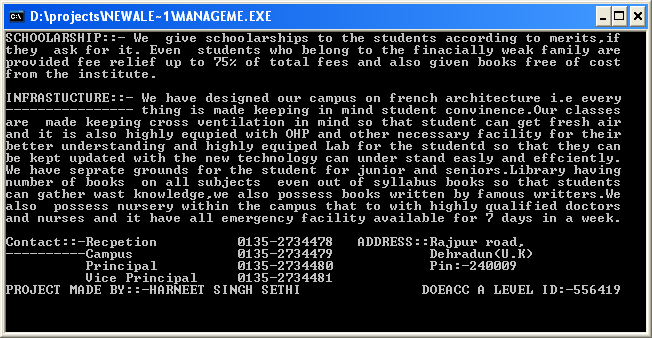
**View Student Marks**

****

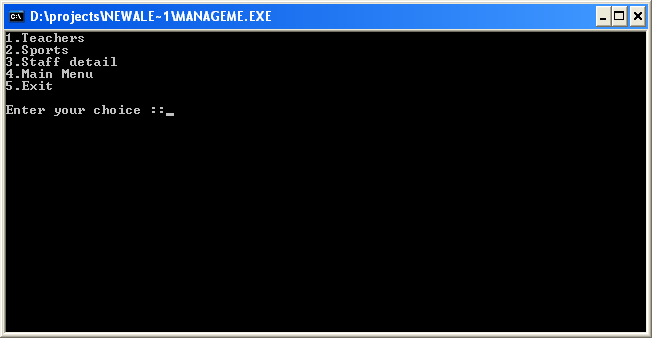
**Class wise result**

****

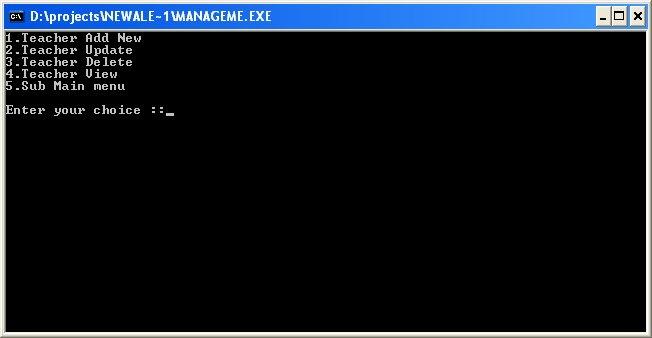
**School History & Information**

****

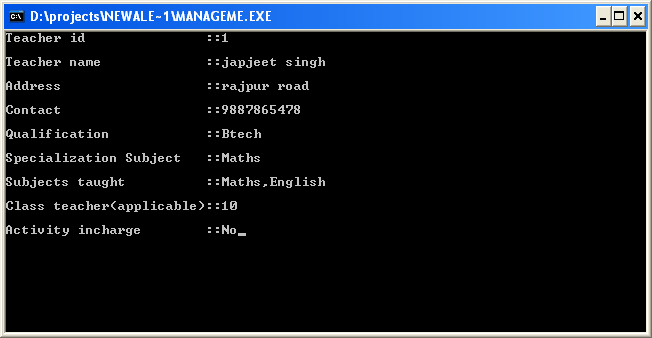
**Miscellaneous Main Menu**

****

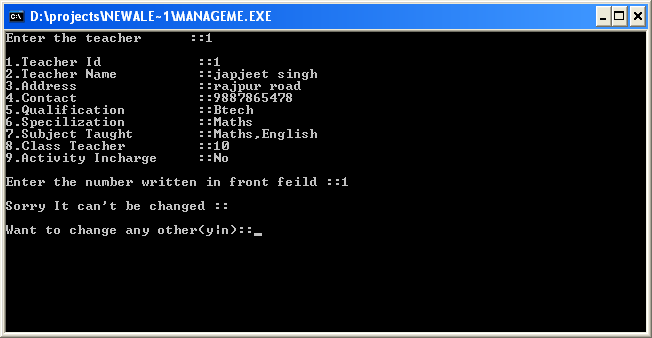
**Teacher Main Menu**

****

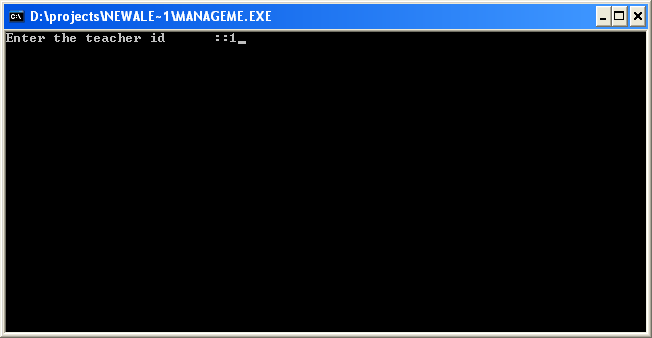
**Teacher Recruitment**

****

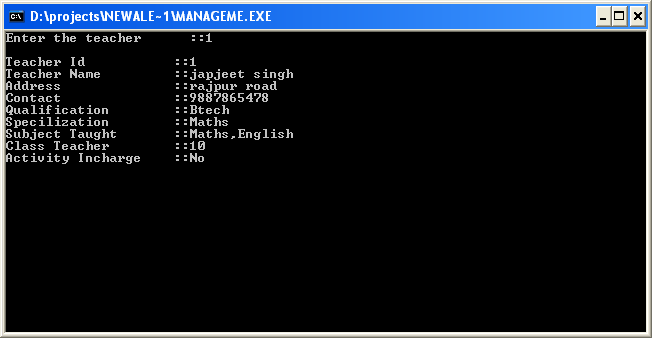
**Update of Teacher**

****

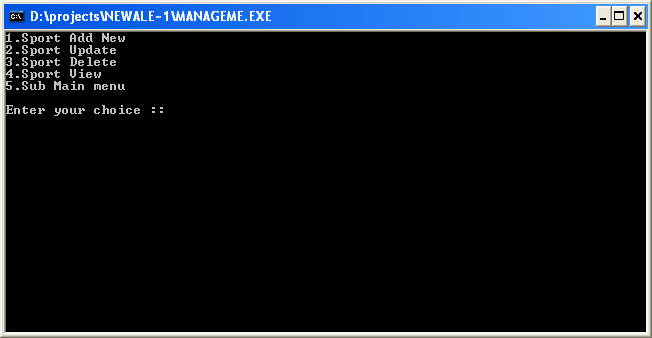
**Deletion of Teacher**

****

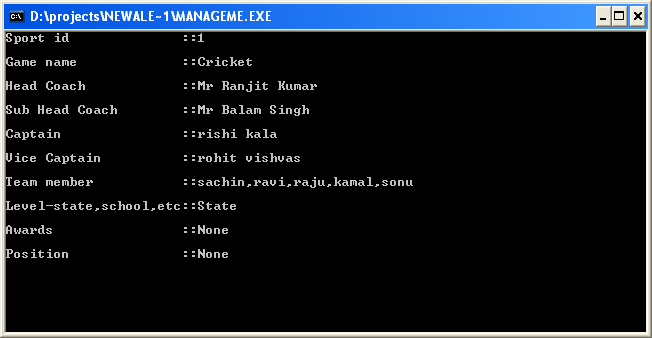
**View of Teacher Detail**

****

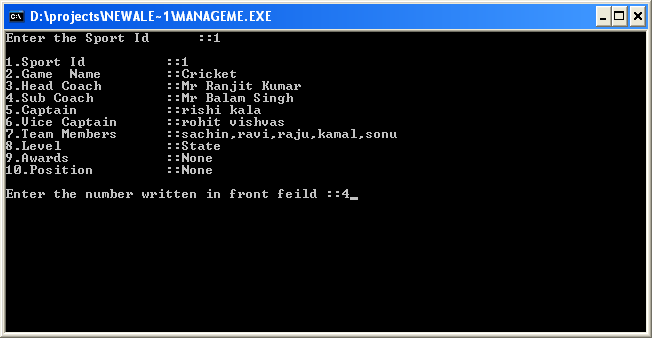
**Sport Main Menu**

****

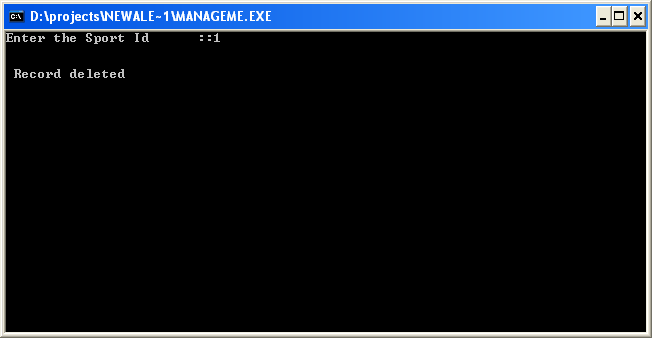
**Sport Detail Added**

****

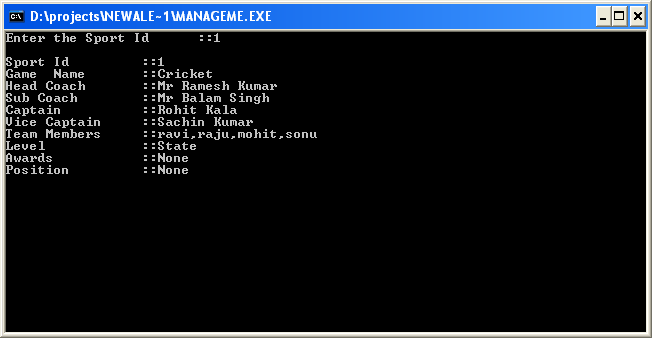
**Update of Sport**

****

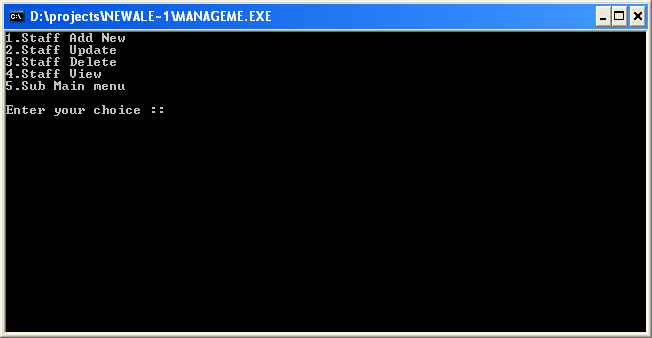
**Deletion of Sport**

****

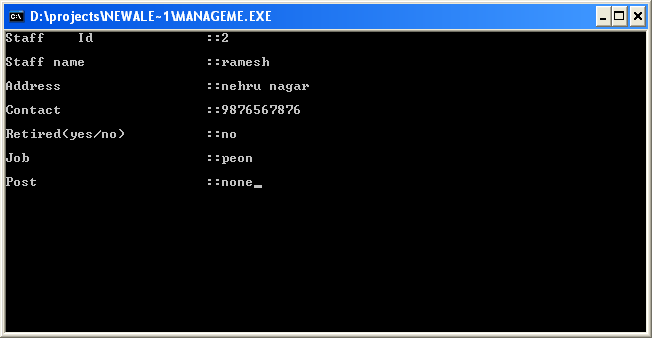
**View of Sport Details**

****

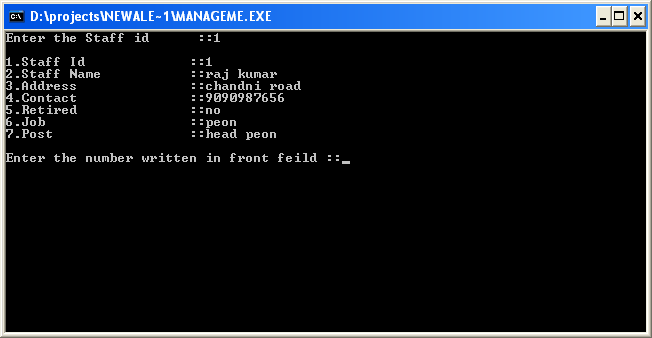
**Staff Main Menu**

****

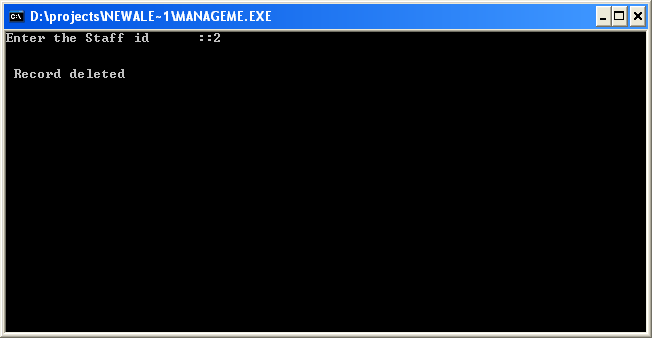
**Staff Member Added**

****

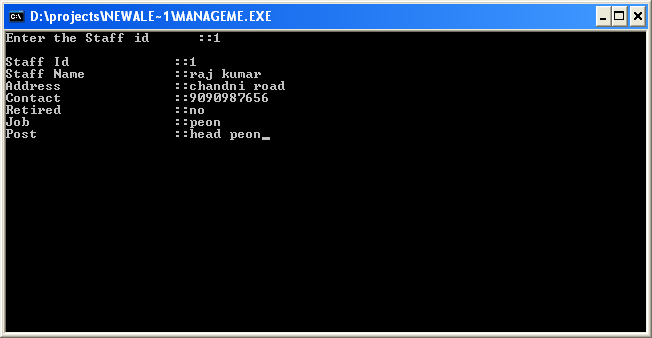
**Update of Staff Detail**

****

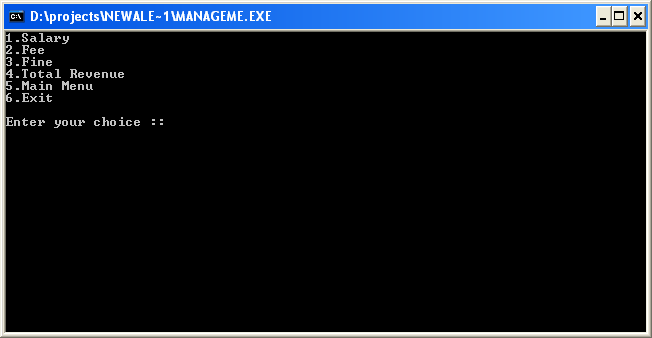
**Deletion of Staff Member**

****

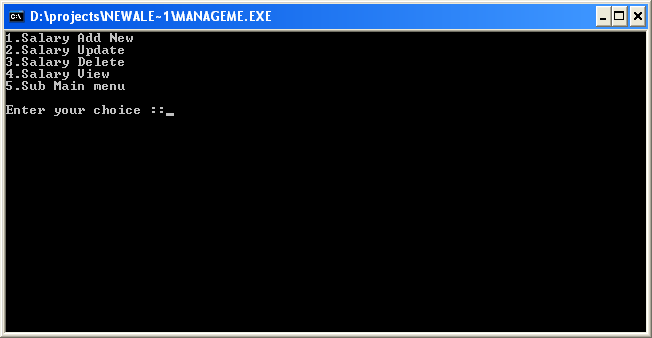
**View of Staff Detail**

****

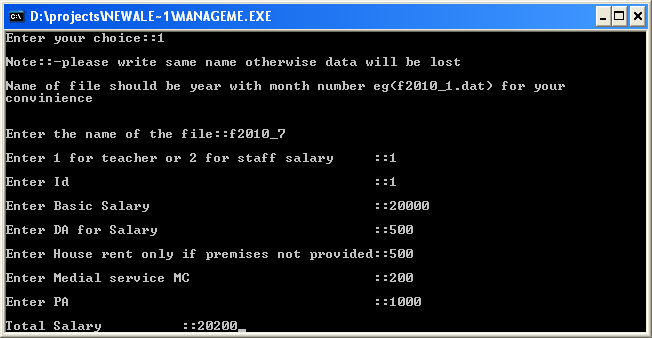
**Account Main menu**

****

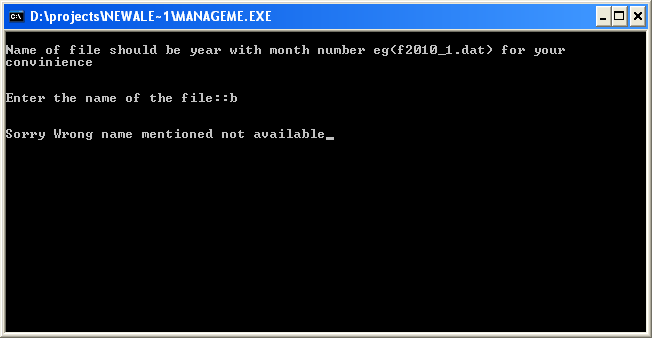
**Salary Main Menu**

****

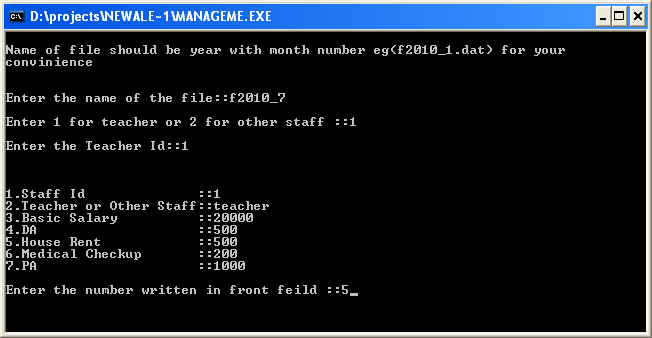
**Enter of Salary Detail**

****

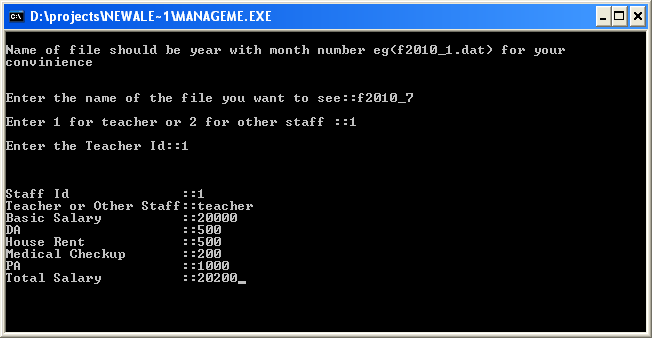
**Update of Salary**

****

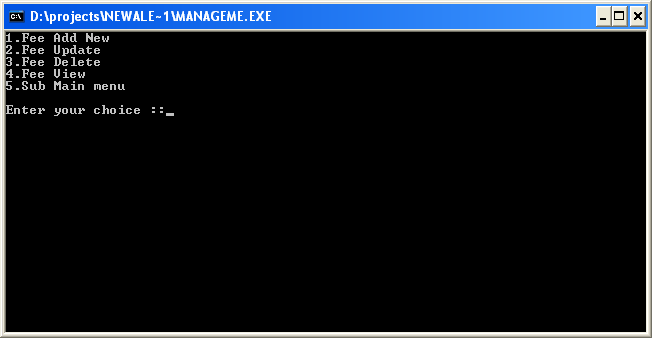
**Deletion of Salary**

****

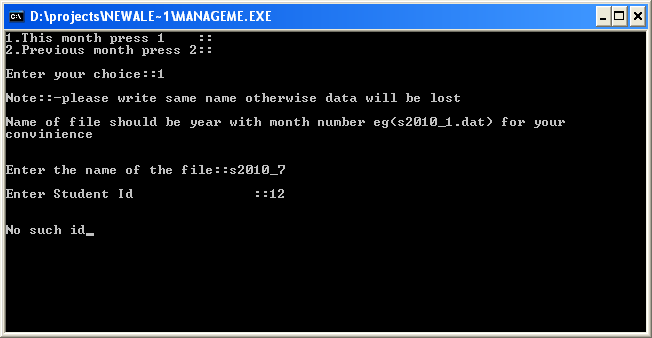
**View of Salary Detail**

****

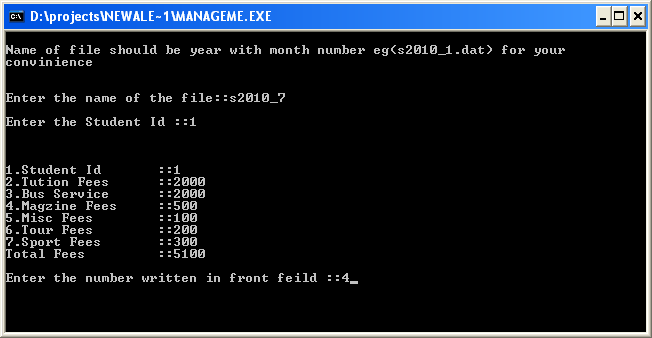
**Fee Main menu**

****

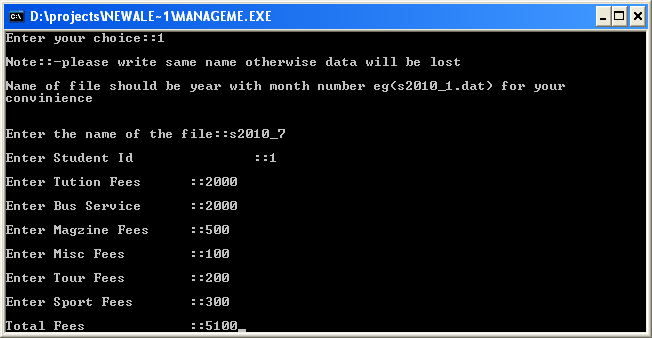
**Enter of Fee Detail**

****

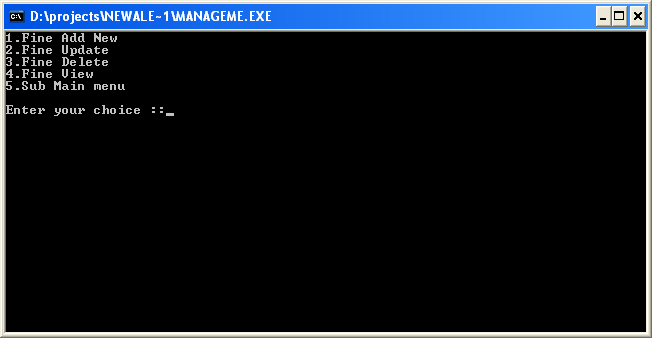
**Update of Fee Detail**

****

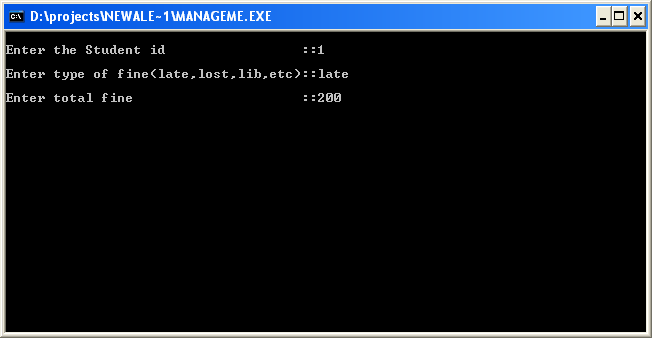
**View of Fee Detail**

****

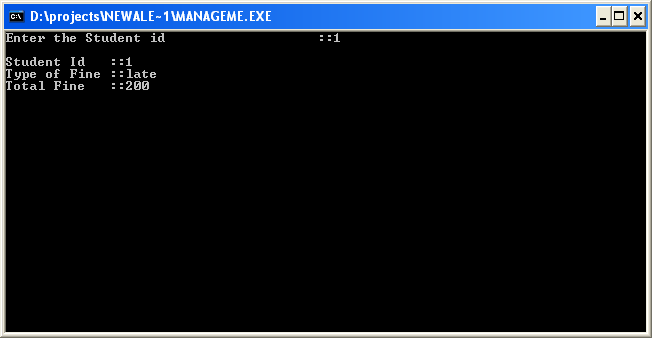
**Fine Main Menu**

****

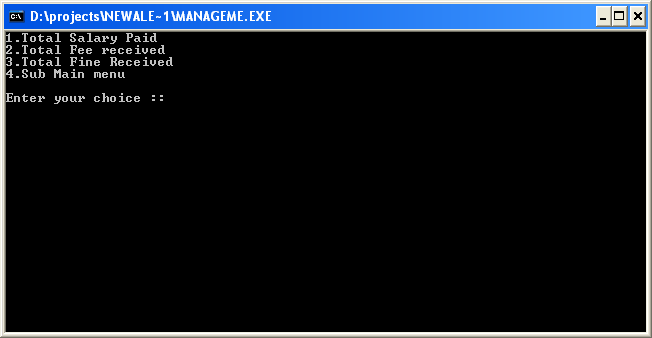
**Insert of Student Fine**

****

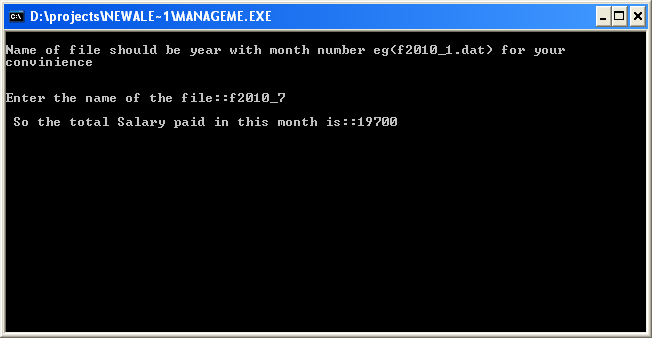
**View of Fine Detail**

****

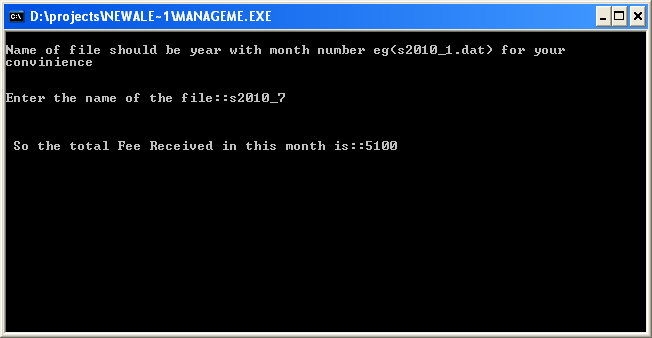
**Revenue Main Menu**

****

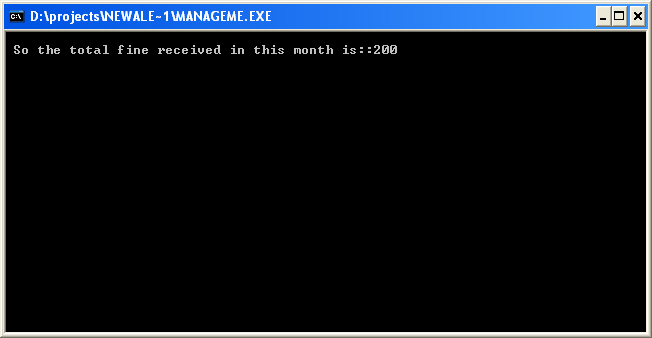
**View of Total Salary Paid**

****

**View of Total Fee Received**

****

**View of Total Fine Received**

****

**SOURCE CODE**

#include<conio.h>

#include<stdio.h>

#include<fstream.h>

#include<string.h>

#include<stdlib.h>

#include<ctype.h>

class student

{

struct trans

{

char name[25];

char char\_certific[5];

int id;

int clas;

char date[10];

};

struct stud

{

char name[25],dob[10];

float father\_income,mother\_income,pert;

int student\_id;

char mother[25],father[25],add[50],contact[25],father\_occp[25],domicile[25],

school\_name[25],tc[5],religion[25],disease[25],blood\_group[5];

};

public:

void admission();

void transfer();

void student\_search();

void student\_submenu();

};

class exam

{

struct mark

{

int mark\_id,clas;

float hindi,eng,geog,phy,chem,bio,com,acc,

comp,eco,hist,social,math,art,science,total;

};

public:

void mark\_exam();

};

class account

{

struct fine

{

int fine\_id;

char type[25];

float fine;

};

struct salary

{

float basic,da,hr,mc;

float pa;

int sal\_id;

char teach[25];

};

struct fee

{

int fee\_id;

float misc,tuition,bus,magzine;

float tour,sport;

};

public:

void salary\_account();

void fee\_account();

void fine\_account();

void revenue\_account();

void account\_submenu();

};

class miscell

{

struct sport

{

int sport\_id;

char game[25],head[25],sub\_head[25],captain[25],vice[25],other[50],level[25],

award[25],position[5];

};

struct teacher

{

int teach\_id;

char name[25],add[50],phone[25],qualific[25],subject[25],taught[25],

clas\_teach[25],activity[25];

};

struct staff

{

int staff\_id;

char name[25],add[50],phone[25],retired[10],job[25],post[25];

};

public:

void misc\_submenu();

void teacher\_member();

void sport\_member();

void staff\_member();

};

class menu

{

static int time;

public:

void start();

void password();

static void times();

};

int menu::time=0;

//------------------------------------------------STUDENT CLASS FUNCTION---

//=============ADMISSION====================================================

void student::admission()

{

stud stu;

while(1)

{

clrscr();

char ans='n';

int last=0,numb=0,id=0,flag=0;

fstream file,file1,temp;

int ch;

cout<<"1.New Student"<<endl<<"2.Student Update"<<endl<<"3.Student Delete"

<<endl<<"4.Sub Main menu";

cout<<"\n\nEnter your choice ::";

cin>>ch;

switch(ch)

{

case 1:

clrscr();

file.open("admission.dat",ios::in | ios::binary);

file.seekg(0,ios::beg);

while(file.read((char\*)&stu,sizeof(stud)))

{

last=1;

id=stu.student\_id;

}

if(last==0)

{

id=1;

}

else

{

id=id+1;

}

file.close();

file1.open("admission.dat",ios::app | ios::binary);

stu.student\_id=id;

cout<<"Student ID ::";

cout<<stu.student\_id;

cout<<"\nStudent Name ::";

fflush(stdin);

gets(stu.name);

cout<<"\nFather Name ::";

fflush(stdin);

gets(stu.father);

cout<<"\nMother Name ::";

fflush(stdin);

gets(stu.mother);

cout<<"\nAddress ::";

fflush(stdin);

gets(stu.add);

cout<<"\nContact No. Parent/Guardian::";

fflush(stdin);

gets(stu.contact);

cout<<"\nFather Occupation ::";

fflush(stdin);

gets(stu.father\_occp);

cout<<"\nFather Income ::";

fflush(stdin);

cin>>stu.father\_income;

cout<<"\nMother Income ::";

fflush(stdin);

cin>>stu.mother\_income;

cout<<"\nBirth certificate /state ::";

fflush(stdin);

gets(stu.domicile);

cout<<"\nOld School ::";

fflush(stdin);

gets(stu.school\_name);

cout<<"\nMarks/Percentage ::";

fflush(stdin);

cin>>stu.pert;

cout<<"\nTC (taken from school)[y|n]::";

fflush(stdin);

gets(stu.tc);

cout<<"\nReligion ::";

fflush(stdin);

gets(stu.religion);

cout<<"\nDisease student suffering ::";

fflush(stdin);

gets(stu.disease);

cout<<"\nBlood group ::";

fflush(stdin);

gets(stu.blood\_group);

cout<<"\nDate of birth [dd/mm/yyyy] ::";

gets(stu.dob);

file1.write((char\*)&stu,sizeof(stud));

file1.close();

break;

case 2:

clrscr();

cout<<"Enter the student Identity Number:: ";

cin>>id;

file.open("admission.dat",ios::in | ios::binary);

file1.open("temp.dat",ios::out | ios::binary);

file.seekg(0,ios::beg);

while(file.read((char\*)&stu,sizeof(stud)))

{

if(stu.student\_id==id)

{

flag=1;

cout<<endl<<"1.Student Id ::"<<stu.student\_id;

cout<<endl<<"2.Student Name ::"<<stu.name;

cout<<endl<<"3.Date of Birth ::"<<stu.dob;

cout<<endl<<"4.Father Name ::"<<stu.father;

cout<<endl<<"5.Mother Name ::"<<stu.mother;

cout<<endl<<"6.Address ::"<<stu.add;

cout<<endl<<"7.Contact ::"<<stu.contact;

cout<<endl<<"8.Father Occupation ::"<<stu.father\_occp;

cout<<endl<<"9.Father Income ::"<<stu.father\_income;

cout<<endl<<"10.Mother Income ::"<<stu.mother\_income;

cout<<endl<<"11.State ::"<<stu.domicile;

cout<<endl<<"12.Old School Name ::"<<stu.school\_name;

cout<<endl<<"13.Percentage ::"<<stu.pert;

cout<<endl<<"14.Transfer(still studying in school)::"<<stu.tc;

cout<<endl<<"15.Religion ::"<<stu.religion;

cout<<endl<<"16.Disease ::"<<stu.disease;

cout<<endl<<"17.Blood Group ::"<<stu.blood\_group;

do

{

cout<<"\n\nEnter the number written in front feild ::";

fflush(stdin);

cin>>numb;

if(numb==1)

{

cout<<endl<<"Sorry It can't be changed ::";

}

else if(numb==2)

{

cout<<endl<<"New Student Name ::";

gets(stu.name);

}

else if(numb==3)

{

cout<<endl<<"New Date Of Birth ::";

gets(stu.dob);

}

else if(numb==4)

{

cout<<endl<<"New Father's Name ::";

gets(stu.father);

}

else if(numb==5)

{

cout<<endl<<"New Mother's Name ::";

gets(stu.mother);

}

else if(numb==6)

{

cout<<endl<<"New Address ::";

gets(stu.add);

}

else if(numb==7)

{

cout<<endl<<"New Contact ::";

gets(stu.contact);

}

else if(numb==8)

{

cout<<endl<<"New Father Occupation ::";

gets(stu.father\_occp);

}

else if(numb==9)

{

cout<<endl<<"New Father Income ::";

cin>>stu.father\_income;

}

else if(numb==10)

{

cout<<endl<<"New Mother Income ::";

cin>>stu.mother\_income;

}

else if(numb==11)

{

cout<<endl<<"New Domicile ::";

gets(stu.domicile);

}

else if(numb==12)

{

cout<<endl<<"New Old School Name ::";

gets(stu.school\_name);

}

else if(numb==13)

{

cout<<endl<<"New Percentage ::";

cin>>stu.pert;

}

else if(numb==14)

{

cout<<endl<<"Sorry cant'n be changed ::";

cout<<endl<<"Give student TC or Expel ::";

}

else if(numb==15)

{

cout<<endl<<"New Religion ::";

gets(stu.religion);

}

else if(numb==16)

{

cout<<endl<<"New Disease ::";

gets(stu.disease);

}

else if(numb==17)

{

cout<<endl<<"New Blood group ::";

gets(stu.blood\_group);

}

else

{

cout<<"\n\nWrong Number choice";

}

cout<<"\n\nWant to change any other(y|n)::";

ans=getche();

}while(ans=='y');

file1.write((char\*)&stu,sizeof(stud));

}

else

{

file1.write((char\*)&stu,sizeof(stud));

}

}

file.close();

file1.close();

if(flag==0)

{

cout<<"Record not found";

getch();

}

else

{

file.open("temp.dat",ios::in | ios::binary);

file1.open("admission.dat",ios::out | ios::binary);

file.seekg(0,ios::beg);

while(file.read((char\*)&stu,sizeof(stud)))

{

file1.write((char\*)&stu,sizeof(stud));

}

file.close();

file1.close();

remove("temp.dat");

}

break;

case 3:

clrscr();

file.open("admission.dat",ios::in | ios::binary);

file1.open("temp.dat",ios::out | ios::binary);

file.seekg(0,ios::beg);

cout<<"Enter the student Identity Number:: ";

cin>>id;

while(file.read((char\*)&stu,sizeof(stud)))

{

if(stu.student\_id==id)

{

cout<<"Record removed";

flag=1;

}

else

{

file1.write((char\*)&stu,sizeof(stud));

}

}

file.close();

file1.close();

if(flag==0)

{

cout<<"Record not found";

}

else

{

file.open("admission.dat",ios::out|ios::binary);

temp.open("temp.dat",ios::in|ios::binary);

temp.seekg(0,ios::beg);

while(temp.read((char\*)&stu,sizeof(stud)))

{

file.write((char\*)&stu,sizeof(stud));

}

file.close();

temp.close();

remove("temp.dat");

}

getch();

break;

case 4:

this->student\_submenu();

break;

default:

cout<<"\nWrong Choice";

getch();

break;

}

}

}

//=====================TC ADMISSION========================================

void student::transfer()

{

stud stu;

trans obj;

while(1)

{

clrscr();

char ans='n';

int numb=0,id=0,flag=0,already=0;

fstream file,file2,temp1,file1,temp;

int ch;

cout<<"1.Student TC"<<endl<<"2.TC Update"<<endl<<"3.TC Delete"

<<endl<<"4.TC View"<<endl<<"5.Sub Main menu";

cout<<"\n\nEnter your choice ::";

cin>>ch;

switch(ch)

{

case 1:

clrscr();

file1.open("admission.dat",ios::in | ios::binary);

cout<<"Enter Student ID ::";

cin>>id;

while(file1.read((char\*)&stu,sizeof(stud)))

{

if(id==stu.student\_id)

{

flag=1;

if(strcmp(stu.tc,"y")==0)

{

already=1;

}

}

}

file1.close();

if(already==0)

{

obj.id=id;

file.open("tc.dat",ios::app | ios::binary);

if(flag==1)

{

cout<<"Enter Student Name ::";

gets(obj.name);

cout<<"Class ::";

cin>>obj.clas;

cout<<"Character Certificate [y/n]::";

gets(obj.char\_certific);

cout<<"Date [dd/mm/yyyy] ::";

gets(obj.date);

file.write((char\*)&obj,sizeof(trans));

file.close();

file.open("admission.dat",ios::in | ios::binary);

file1.open("temp.dat",ios::out | ios::binary);

file.seekg(0,ios::beg);

while(file.read((char\*)&stu,sizeof(stud)))

{

if(stu.student\_id==id)

{

strcpy(stu.tc,"y");

file1.write((char\*)&stu,sizeof(stud));

}

else

{

file1.write((char\*)&stu,sizeof(stud));

}

}

file.close();

file1.close();

file.open("admission.dat",ios::out|ios::binary);

temp.open("temp.dat",ios::in|ios::binary);

temp.seekg(0,ios::beg);

while(temp.read((char\*)&stu,sizeof(stud)))

{

file.write((char\*)&stu,sizeof(stud));

}

file.close();

temp.close();

remove("temp.dat");

}

else

{

cout<<"\n\n NO SUCH ID";

getch();

}

}

else

{

cout<<"\n\n Already TC or Expeled ";

getch();

}

break;

case 2:

clrscr();

file.open("tc.dat", ios::in | ios::binary);

file1.open("temp.dat",ios::out | ios::binary);

file.seekg(0,ios::beg);

cout<<"Enter the Student id ::";

cin>>id;

while(file.read((char\*)&obj,sizeof(trans)))

{

if(id==obj.id)

{

flag=1;

cout<<endl<<"1.Student Id ::"<<obj.id;

cout<<endl<<"2.Student Name ::"<<obj.name;

cout<<endl<<"3.Class ::"<<obj.clas;

cout<<endl<<"4.Date of TC(given) ::"<<obj.date;

cout<<endl<<"5.Character certificate::"<<obj.char\_certific;

do

{

cout<<"\n\nEnter the number written in front feild ::";

fflush(stdin);

cin>>numb;

if(numb==1)

{

cout<<endl<<"Sorry It can't be changed ::";

}

else if(numb==2)

{

cout<<endl<<"New Student Name ::";

gets(obj.name);

}

else if(numb==3)

{

cout<<endl<<"New Class ::";

cin>>obj.clas;

}

else if(numb==4)

{

cout<<endl<<"New Date Of TC ::";

gets(obj.date);

}

else if(numb==5)

{

cout<<endl<<"New Character certificate given (y/n)::";

gets(obj.char\_certific);

}

else

{

cout<<"\n\nWrong Number choice";

}

cout<<"\n\nWant to change any other(y|n)::";

ans=getche();

}while(ans=='y');

file1.write((char\*)&obj,sizeof(trans));

}

else

{

file1.write((char\*)&obj,sizeof(trans));

}

}

file.close();

file1.close();

if(flag==0)

{

cout<<"Record not found";

getch();

}

else

{

file.open("temp.dat",ios::in | ios::binary);

file1.open("tc.dat",ios::out | ios::binary);

file.seekg(0,ios::beg);

while(file.read((char\*)&obj,sizeof(trans)))

{

file1.write((char\*)&obj,sizeof(trans));

}

file.close();

file1.close();

remove("temp.dat");

}

break;

case 3:

clrscr();

file.open("tc.dat", ios::in | ios::binary);

file1.open("temp.dat",ios::out | ios::binary);

file.seekg(0,ios::beg);

cout<<"Enter the Student id ::";

cin>>id;

while(file.read((char\*)&obj,sizeof(trans)))

{

if(id==obj.id)

{

flag=1;

cout<<"\n\n Record deleted";

file2.open("admission.dat",ios::in | ios::binary);

temp.open("temp1.dat",ios::out | ios::binary);

while(file2.read((char\*)&stu,sizeof(stud)))

{

if(id==stu.student\_id)

{

strcpy(stu.tc,"n");

temp.write((char\*)&stu,sizeof(stud));

}

else

{

temp.write((char\*)&stu,sizeof(stud));

}

}

}

else

{

file1.write((char\*)&obj,sizeof(trans));

}

}

file.close();

file1.close();

file2.close();

temp.close();

if(flag==0)

{

cout<<"Record not found";

getch();

}

else

{

file.open("temp.dat",ios::in | ios::binary);

file1.open("tc.dat",ios::out | ios::binary);

file.seekg(0,ios::beg);

while(file.read((char\*)&obj,sizeof(trans)))

{

file1.write((char\*)&obj,sizeof(trans));

}

file.close();

file1.close();

file2.open("admission.dat",ios::out | ios::binary);

temp.open("temp1.dat",ios::in | ios::binary);

while(temp.read((char\*)&stu,sizeof(stud)))

{

file2.write((char\*)&stu,sizeof(stud));

}

file2.close();

temp.close();

remove("temp1.dat");

remove("temp.dat");

}

getch();

break;

case 4:

clrscr();

file.open("tc.dat", ios::in | ios::binary);

file.seekg(0,ios::beg);

cout<<"Enter the Student id ::";

cin>>id;

while(file.read((char\*)&obj,sizeof(trans)))

{

if(id==obj.id)

{

flag=1;

cout<<endl<<"Student Id ::"<<obj.id;

cout<<endl<<"Student Name ::"<<obj.name;

cout<<endl<<"Class ::"<<obj.clas;

cout<<endl<<"Date of TC(given) ::"<<obj.date;

cout<<endl<<"Character certificate::"<<obj.char\_certific;

}

}

file.close();

if(flag==0)

{

cout<<"\n\nRecord not found";

}

getch();

break;

case 5:

this->student\_submenu();

break;

default:

cout<<"\n\nWrong Choice::";

break;

}

}

}

//=================STUDENT SEARCH==========================================

void student::student\_search()

{

stud stu;

fstream file;

int id=0,flag=0;

clrscr();

file.open("admission.dat",ios::in | ios::binary);

file.seekg(0,ios::beg);

cout<<"Enter the student Identity Number::";

cin>>id;

while(file.read((char\*)&stu,sizeof(stud)))

{

if(stu.student\_id==id)

{

flag=1;

cout<<endl<<"Student Id ::"<<stu.student\_id;

cout<<endl<<"Student Name ::"<<stu.name;

cout<<endl<<"Date of Birth ::"<<stu.dob;

cout<<endl<<"Father Name ::"<<stu.father;

cout<<endl<<"Mother Name ::"<<stu.mother;

cout<<endl<<"Address Parent ::"<<stu.add;

cout<<endl<<"Contact ::"<<stu.contact;

cout<<endl<<"Father Occupation ::"<<stu.father\_occp;

cout<<endl<<"Father Income ::"<<stu.father\_income;

cout<<endl<<"Mother Income ::"<<stu.mother\_income;

cout<<endl<<"State ::"<<stu.domicile;

cout<<endl<<"Old School Name ::"<<stu.school\_name;

cout<<endl<<"Percentage ::"<<stu.pert;

cout<<endl<<"Transfer ::"<<stu.tc;

cout<<endl<<"Religion ::"<<stu.religion;

cout<<endl<<"Disease ::"<<stu.disease;

cout<<endl<<"Blood Group ::"<<stu.blood\_group;

}

}

if(flag==0)

{

cout<<"No such record";

}

file.close();

}

//------------------------------------------------EXAMINATION CLASS----------

//====================MARKS==================================================

void exam::mark\_exam()

{

mark mar;

while(1)

{

clrscr();

char ans='n',filename[25];

int id=0,numb=0,flag=0,last=0,ch=0,choice=0;

fstream file,temp,file1;

cout<<"1.Enter Student Marks"<<endl<<"2.Enter Student Marks Update"<<endl

<<"3.Student Marks Delete"<<endl<<"4.Student Marks View"<<endl

<<"5.View Class Marks"<<endl<<"6.Sub Main menu";

cout<<"\n\nEnter your choice ::";

cin>>ch;

switch(ch)

{

case 1:

clrscr();

student::stud st;

cout<<"\nNote::-please write same name otherwise data will be lost\n";

cout<<"\nName of file should be year with class number eg(c2010\_1.dat) for your";

cout<<"\nconvinience";

cout<<"\n\n\nEnter the name of the file::";

gets(filename);

cout<<"\nEnter the class::";

cin>>choice;

if(choice>=1 && choice<6)

{

do

{

clrscr();

flag=0;

last=0;

cout<<"\nEnter Student Id ::";

cin>>id;

file1.open("admission.dat",ios::in | ios::binary);

while(file1.read((char\*)&st,sizeof(student::stud)))

{

if(id==st.student\_id)

{

flag=1;

}

}

file1.close();

if(flag==1)

{

temp.open(filename,ios::in |ios::binary);

temp.seekg(0,ios::beg);

while(temp.read((char\*)&mar,sizeof(exam::mark)))

{

if(id==mar.mark\_id)

{

last=1;

}

}

temp.close();

if(last==0)

{

file.open(filename,ios::app | ios::binary);

mar.mark\_id=id;

mar.clas=choice;

cout<<"\nEnter Hindi Marks ::";

cin>>mar.hindi;

cout<<"\nEnter English Marks ::";

cin>>mar.eng;

cout<<"\nEnter Science Marks ::";

cin>>mar.science;

cout<<"\nEnter Social Marks ::";

cin>>mar.social;

cout<<"\nEnter Maths Marks ::";

cin>>mar.math;

cout<<"\nTotal Marks ::";

mar.total=mar.hindi+mar.eng+mar.science+mar.social

+mar.math;

cout<<mar.total;

mar.geog=0,mar.phy=0,mar.chem=0,mar.bio=0,mar.com=0,

mar.eco=0,mar.hist=0,mar.art=0,mar.comp=0,mar.acc=0;

file.write((char\*)&mar,sizeof(exam::mark));

file.close();

}

else

{

cout<<"\n\n Already Exist";

}

}

else

{

cout<<"\n\n NO SUCH ID";

}

cout<<"\n\n Want to enter more marks::";

ans=getche();

}while(ans=='y');

}

else if(choice>=6 && choice<9)

{

do

{

clrscr();

flag=0;

last=0;

cout<<"\nEnter Student Id ::";

cin>>id;

file1.open("admission.dat",ios::in | ios::binary);

while(file1.read((char\*)&st,sizeof(student::stud)))

{

cout<<st.student\_id;

if(id==st.student\_id)

{

flag=1;

}

}

file1.close();

if(flag==1)

{

temp.open(filename,ios::in |ios::binary);

temp.seekg(0,ios::beg);

while(temp.read((char\*)&mar,sizeof(exam::mark)))

{

if(id==mar.mark\_id)

{

last=1;

}

}

temp.close();

if(last==1)

{

file.open(filename,ios::app | ios::binary);

mar.mark\_id=id;

mar.clas=choice;

cout<<"\nEnter Hindi Marks ::";

cin>>mar.hindi;

cout<<"\nEnter English Marks ::";

cin>>mar.eng;

cout<<"\nEnter Physics Marks ::";

cin>>mar.phy;

cout<<"\nEnter Chemistry Marks ::";

cin>>mar.chem;

cout<<"\nEnter Biology Marks ::";

cin>>mar.bio;

cout<<"\nEnter History Marks ::";

cin>>mar.hist;

cout<<"\nEnter Geography Marks ::";

cin>>mar.geog;

cout<<"\nEnter Maths Marks ::";

cin>>mar.math;

cout<<"\nTotal Marks ::";

mar.total=mar.hindi+mar.eng+mar.phy+mar.chem

+mar.bio+mar.hist+mar.geog+mar.math;

cout<<mar.total;

mar.com=0,mar.eco=0,mar.hist=0,mar.art=0,mar.comp=0

,mar.acc=0;

file.write((char\*)&mar,sizeof(exam::mark));

file.close();

}

else

{

cout<<"\n\n Already Exist";

}

}

else

{

cout<<"\n\n NO SUCH ID";

}

cout<<"\n\n Want to enter more marks::";

ans=getche();

}while(ans=='y');

}

else if(choice>=9 && choice<11)

{

cout<<"\n1.COMPUTER CLASS";

cout<<"\n2.ECONOMICS CLASS";

cout<<"\n\nEnter your choice";

cin>>numb;

if(numb==1)

{

do

{

clrscr();

flag=0;

last=0;

cout<<"\nEnter Student Id ::";

cin>>id;

file1.open("admission.dat",ios::in | ios::binary);

while(file1.read((char\*)&st,sizeof(student::stud)))

{

cout<<st.student\_id;

if(id==st.student\_id)

{

flag=1;

}

}

file1.close();

if(flag==1)

{

temp.open(filename,ios::in |ios::binary);

temp.seekg(0,ios::beg);

while(temp.read((char\*)&mar,sizeof(exam::mark)))

{

if(id==mar.mark\_id)

{

last=1;

}

}

temp.close();

if(last==0)

{

file.open(filename,ios::app | ios::binary);

mar.mark\_id=id;

mar.clas=choice;

cout<<"\nEnter Hindi Marks ::";

cin>>mar.hindi;

cout<<"\nEnter English Marks ::";

cin>>mar.eng;

cout<<"\nEnter Physics Marks ::";

cin>>mar.phy;

cout<<"\nEnter Chemistry Marks ::";

cin>>mar.chem;

cout<<"\nEnter Biology Marks ::";

cin>>mar.bio;

cout<<"\nEnter History Marks ::";

cin>>mar.hist;

cout<<"\nEnter Geography Marks ::";

cin>>mar.geog;

cout<<"\nEnter Maths Marks ::";

cin>>mar.math;

cout<<"Enter Computer Marks ::";

cin>>mar.comp;

cout<<"\nTotal Marks ::";

mar.total=mar.hindi+mar.eng+mar.phy+mar.chem

+mar.bio+mar.hist+mar.geog+mar.math+mar.comp;

cout<<mar.total;

mar.com=0,mar.eco=0,mar.hist=0,mar.art=0,mar.acc=0;

file.write((char\*)&mar,sizeof(exam::mark));

file.close();

}

else

{

cout<<"\n\n Already Exist";

}

}

else

{

cout<<"\n\n NO SUCH ID";

}

cout<<"\n\n Want to enter more marks::";

ans=getche();

}while(ans=='y');

}

else if(numb==2)

{

do

{

clrscr();

flag=0;

last=0;

cout<<"\nEnter Student Id ::";

cin>>id;

file1.open("admission.dat",ios::in | ios::binary);

while(file1.read((char\*)&st,sizeof(student::stud)))

{

cout<<st.student\_id;

if(id==st.student\_id)

{

flag=1;

}

}

file1.close();

if(flag==1)

{

temp.open(filename,ios::in |ios::binary);

temp.seekg(0,ios::beg);

while(temp.read((char\*)&mar,sizeof(exam::mark)))

{

if(id==mar.mark\_id)

{

last=1;

}

}

temp.close();

if(last==0)

{

file.open(filename,ios::app | ios::binary);

mar.mark\_id=id;

mar.clas=choice;

cout<<"\nEnter Hindi Marks ::";

cin>>mar.hindi;

cout<<"\nEnter English Marks ::";

cin>>mar.eng;

cout<<"\nEnter Physics Marks ::";

cin>>mar.phy;

cout<<"\nEnter Chemistry Marks ::";

cin>>mar.chem;

cout<<"\nEnter Biology Marks ::";

cin>>mar.bio;

cout<<"\nEnter History Marks ::";

cin>>mar.hist;

cout<<"\nEnter Geography Marks ::";

cin>>mar.geog;

cout<<"\nEnter Maths Marks ::";

cin>>mar.math;

cout<<"Enter Economics Marks ::";

cin>>mar.eco;

cout<<"\nTotal Marks ::";

mar.total=mar.hindi+mar.eng+mar.phy+mar.chem

+mar.bio+mar.hist+mar.geog+mar.math+mar.eco;

cout<<mar.total;

mar.com=0,mar.comp=0,mar.hist=0,mar.art=0,mar.acc=0;

file.write((char\*)&mar,sizeof(exam::mark));

file.close();

}

else

{

cout<<"\n\n Already Exist";

}

}

else

{

cout<<"\n\n NO SUCH ID";

}

cout<<"\n\n Want to enter more marks::";

ans=getche();

}while(ans=='y');

}

else

{

cout<<"\n\n\n Wrong choice";

}

}

else if(choice>=11 && choice<=12)

{

cout<<"\nSCIENCE SECTION";

cout<<"\nECONOMICS SECTION";

cout<<"\n\n Enter your choice";

cin>>numb;

if(numb==1)

{

do

{ clrscr();

flag=0;

last=0;

cout<<"\nEnter Student Id ::";

cin>>id;

file1.open("admission.dat",ios::in | ios::binary);

while(file1.read((char\*)&st,sizeof(student::stud)))

{

cout<<st.student\_id;

if(id==st.student\_id)

{

flag=1;

}

}

file1.close();

if(flag==1)

{

temp.open(filename,ios::in |ios::binary);

temp.seekg(0,ios::beg);

while(temp.read((char\*)&mar,sizeof(exam::mark)))

{

if(id==mar.mark\_id)

{

last=1;

}

}

temp.close();

if(last==0)

{

file.open(filename,ios::app | ios::binary);

mar.mark\_id=id;

mar.clas=choice;

cout<<"\nEnter Hindi Marks ::";

cin>>mar.hindi;

cout<<"\nEnter English Marks ::";

cin>>mar.eng;

cout<<"\nEnter Physics Marks ::";

cin>>mar.phy;

cout<<"\nEnter Chemistry Marks ::";

cin>>mar.chem;

cout<<"\nEnter Biology Marks ::";

cin>>mar.bio;

cout<<"\nEnter Computer Marks ::";

cin>>mar.comp;

cout<<"\nEnter Maths Marks ::";

cin>>mar.math;

cout<<"\nTotal Marks ::";

mar.total=mar.hindi+mar.eng+mar.phy+mar.chem+

mar.bio+mar.comp+mar.math;

cout<<mar.total;

mar.geog=0,mar.social=0,mar.com=0,mar.science=0,

mar.eco=0,mar.hist=0,mar.art=0,mar.acc=0;

file.write((char\*)&mar,sizeof(exam::mark));

file.close();

}

else

{

cout<<"\n\n Already Exist";

}

}

else

{

cout<<"\n\n NO SUCH ID";

}

cout<<"\n\n Want to enter more marks::";

ans=getche();

}while(ans=='y');

}

else if(numb==2)

{

do

{

clrscr();

flag=0;

last=0;

cout<<"\nEnter Student Id ::";

cin>>id;

file1.open("admission.dat",ios::in | ios::binary);

while(file1.read((char\*)&st,sizeof(student::stud)))

{

cout<<st.student\_id;

if(id==st.student\_id)

{

flag=1;

}

}

file1.close();

if(flag==1)

{

temp.open(filename,ios::in |ios::binary);

temp.seekg(0,ios::beg);

while(temp.read((char\*)&mar,sizeof(exam::mark)))

{

if(id==mar.mark\_id)

{

last=1;

}

}

temp.close();

if(last==0)

{

file.open(filename,ios::app | ios::binary);

mar.mark\_id=id;

mar.clas=choice;

cout<<"\nEnter Hindi Marks ::";

cin>>mar.hindi;

cout<<"\nEnter English Marks ::";

cin>>mar.eng;

cout<<"\nEnter Commerce Marks ::";

cin>>mar.com;

cout<<"\nEnter Economics Marks ::";

cin>>mar.eco;

cout<<"\nEnter Art Marks ::";

cin>>mar.art;

cout<<"\nEnter Account Marks ::";

cin>>mar.art;

cout<<"\nEnter Computer Marks ::";

cin>>mar.comp;

cout<<"\nEnter Maths Marks ::";

cin>>mar.math;

cout<<"\nTotal Marks ::";

mar.total=mar.hindi+mar.eng+mar.com+mar.eco+

mar.acc+mar.comp+mar.math+mar.art;

cout<<mar.total;

mar.geog=0,mar.social=0,mar.bio=0,mar.chem=0,mar.phy=0

,mar.science=0,mar.hist=0;

file.write((char\*)&mar,sizeof(exam::mark));

file.close();

}

else

{

cout<<"\n\n Already Exist";

}

}

else

{

cout<<"\n\n NO SUCH ID";

}

cout<<"\n\n Want to enter more marks::";

ans=getche();

}while(ans=='y');

}

else

{

cout<<"\n\n Wrong choice";

}

}

else

{

cout<<"\n\nWrong choice";

}

getch();

break;

case 2:

clrscr();

cout<<"\nNote::-please write same name otherwise data will be lost\n";

cout<<"\nName of file should be year with class number eg(c2010\_1.dat) for your";

cout<<"\nconvinience";

cout<<"\n\n\nEnter the name of the file::";

gets(filename);

file.open(filename,ios::nocreate | ios::in | ios::binary);

temp.open("temp.dat",ios::out | ios::binary);

file.seekg(0,ios::beg);

if(!file)

{

cout<<"\nSorry Wrong File name is given";

}

else

{

cout<<"\nEnter the class::";

cin>>choice;

if(choice>=1 && choice<6)

{

cout<<"\nEnter Student Id ::";

cin>>id;

while(file.read((char\*)&mar,sizeof(exam::mark)))

{

if(id==mar.mark\_id)

{

flag=1;

cout<<"\n1.Student ID ::"<<mar.mark\_id;

cout<<"\n2.Student Class ::"<<mar.clas;

cout<<"\n3.Student Hindi Marks ::"<<mar.hindi;

cout<<"\n4.Student English Marks ::"<<mar.eng;

cout<<"\n5.Student Science Marks ::"<<mar.science;

cout<<"\n6.Student Social Marks ::"<<mar.social;

cout<<"\n7.Student Maths Marks ::"<<mar.math;

cout<<"\nTotal Marks ::";

mar.total=mar.hindi+mar.eng+mar.science+mar.social

+mar.math;

cout<<mar.total;

mar.geog=0,mar.phy=0,mar.chem=0,mar.bio=0,mar.com=0,

mar.eco=0,mar.hist=0,mar.art=0,mar.comp=0,mar.acc=0;

do

{

cout<<"\n\nEnter the number written in front feild ::";

cin>>numb;

if(numb==1)

{

cout<<endl<<"Sorry It can't be changed ::";

}

else if(numb==2)

{

cout<<endl<<"Sorry It can't be changed ::";

}

else if(numb==3)

{

cout<<endl<<"New Hindi Marks ::";

cin>>mar.hindi;

}

else if(numb==4)

{

cout<<endl<<"New English Marks ::";

cin>>mar.eng;

}

else if(numb==5)

{

cout<<endl<<"New Science Marks ::";

cin>>mar.science;

}

else if(numb==6)

{

cout<<endl<<"New Social Marks ::";

cin>>mar.social;

}

else if(numb==7)

{

cout<<endl<<"New Maths Marks ::";

cin>>mar.math;

}

else

{

cout<<"\n\nWrong Number choice";

}

cout<<"\n\nWant to change any other(y|n)::";

ans=getche();;

}while(ans=='y');

temp.write((char\*)&mar,sizeof(exam::mark));

}

else

{

temp.write((char\*)&mar,sizeof(exam::mark));

}

}

file.close();

temp.close();

}

else if(choice>=6 && choice<9)

{

cout<<"\nEnter Student Id ::";

cin>>id;

while(file.read((char\*)&mar,sizeof(exam::mark)))

{

if(id==mar.mark\_id)

{

flag=1;

cout<<"\n1.Student ID ::"<<mar.mark\_id;

cout<<"\n2.Student Class ::"<<mar.clas;

cout<<"\n3.Student Hindi Marks ::"<<mar.hindi;

cout<<"\n4.Student English Marks ::"<<mar.eng;

cout<<"\n5.Student Physics Marks ::"<<mar.phy;

cout<<"\n6.Student Chemistry Marks ::"<<mar.chem;

cout<<"\n7.Student Biology Marks ::"<<mar.bio;

cout<<"\n8.Student History Marks ::"<<mar.hist;

cout<<"\n9.Student Geography Marks ::"<<mar.geog;

cout<<"\n10.Student Maths Marks ::"<<mar.math;

cout<<"\nTotal Marks ::";

mar.total=mar.hindi+mar.eng+mar.phy+mar.chem

+mar.bio+mar.hist+mar.geog+mar.math;

cout<<mar.total;

mar.com=0,mar.eco=0,mar.hist=0,mar.art=0,mar.comp=0,mar.acc=0;

do

{

cout<<"\n\nEnter the number written in front feild ::";

cin>>numb;

if(numb==1)

{

cout<<endl<<"Sorry It can't be changed ::";

}

else if(numb==2)

{

cout<<endl<<"Sorry It can't be changed ::";

}

else if(numb==3)

{

cout<<endl<<"New Hindi Marks ::";

cin>>mar.hindi;

}

else if(numb==4)

{

cout<<endl<<"New English Marks ::";

cin>>mar.eng;

}

else if(numb==5)

{

cout<<endl<<"New Physics Marks ::";

cin>>mar.phy;

}

else if(numb==6)

{

cout<<endl<<"New Chemistry Marks ::";

cin>>mar.chem;

}

else if(numb==7)

{

cout<<endl<<"New Biology Marks ::";

cin>>mar.bio;

}

else if(numb==8)

{

cout<<endl<<"New History Marks ::";

cin>>mar.hist;

}

else if(numb==9)

{

cout<<endl<<"New Geography Marks ::";

cin>>mar.geog;

}

else if(numb==10)

{

cout<<endl<<"New Maths Marks ::";

cin>>mar.math;

}

else

{cout<<"\n\nWrong Number choice";

}

cout<<"\n\nWant to change any other(y|n)::";

ans=getche();;

}while(ans=='y');

temp.write((char\*)&mar,sizeof(exam::mark));

}

else

{

temp.write((char\*)&mar,sizeof(exam::mark));

}

}

file.close();

temp.close();

}

else if(choice>=9 && choice<11)

{

cout<<"\n1.COMPUTER CLASS";

cout<<"\n2.ECONOMICS CLASS";

cout<<"\n\nEnter your choice";

cin>>last;

if(last==1)

{

cout<<"\nEnter Student Id ::";

cin>>id;

while(file.read((char\*)&mar,sizeof(exam::mark)))

{

if(id==mar.mark\_id)

{

flag=1;

cout<<"\n1.Student ID ::"<<mar.mark\_id;

cout<<"\n2.Student Class ::"<<mar.clas;

cout<<"\n3.Student Hindi Marks ::"<<mar.hindi;

cout<<"\n4.Student English Marks ::"<<mar.eng;

cout<<"\n5.Student Physics Marks ::"<<mar.phy;

cout<<"\n6.Student Chemistry Marks ::"<<mar.chem;

cout<<"\n7.Student Biology Marks ::"<<mar.bio;

cout<<"\n8.Student History Marks ::"<<mar.hist;

cout<<"\n9.Student Geography Marks ::"<<mar.geog;

cout<<"\n10.Student Maths Marks ::"<<mar.math;

cout<<"\n11.Student Computer Marks ::"<<mar.comp;

cout<<"\nTotal Marks ::";

mar.total=mar.hindi+mar.eng+mar.phy+mar.chem

+mar.bio+mar.hist+mar.geog+mar.math+mar.comp;

cout<<mar.total;

mar.com=0,mar.eco=0,mar.hist=0,mar.art=0,mar.acc=0;

do

{

cout<<"\n\nEnter the number written in front feild ::";

cin>>numb;

if(numb==1)

{

cout<<endl<<"Sorry It can't be changed ::";

}

else if(numb==2)

{

cout<<endl<<"Sorry It can't be changed ::";

}

else if(numb==3)

{

cout<<endl<<"New Hindi Marks ::";

cin>>mar.hindi;

}

else if(numb==4)

{

cout<<endl<<"New English Marks ::";

cin>>mar.eng;

}

else if(numb==5)

{

cout<<endl<<"New Physics Marks ::";

cin>>mar.phy;

}

else if(numb==6)

{

cout<<endl<<"New Chemistry Marks ::";

cin>>mar.chem;

}

else if(numb==7)

{

cout<<endl<<"New Biology Marks ::";

cin>>mar.bio;

}

else if(numb==8)

{

cout<<endl<<"New History Marks ::";

cin>>mar.hist;

}

else if(numb==9)

{

cout<<endl<<"New Geography Marks ::";

cin>>mar.geog;

}

else if(numb==10)

{cout<<endl<<"New Maths Marks ::";

cin>>mar.math;

}

else if(numb==11)

{cout<<endl<<"New Computer Marks ::";

cin>>mar.comp;

}

else

{cout<<"\n\nWrong Number choice";

}

cout<<"\n\nWant to change any other(y|n)::";

ans=getche();;

}while(ans=='y');

temp.write((char\*)&mar,sizeof(exam::mark));

}

else

{

temp.write((char\*)&mar,sizeof(exam::mark));

}

}

file.close();

temp.close();

}

else if(last==2)

{

cout<<"\nEnter Student Id ::";

cin>>id;

while(file.read((char\*)&mar,sizeof(exam::mark)))

{

if(id==mar.mark\_id)

{

flag=1;

cout<<"\n1.Student ID ::"<<mar.mark\_id;

cout<<"\n2.Student Class ::"<<mar.clas;

cout<<"\n3.Student Hindi Marks ::"<<mar.hindi;

cout<<"\n4.Student English Marks ::"<<mar.eng;

cout<<"\n5.Student Physics Marks ::"<<mar.phy;

cout<<"\n6.Student Chemistry Marks ::"<<mar.chem;

cout<<"\n7.Student Biology Marks ::"<<mar.bio;

cout<<"\n8.Student History Marks ::"<<mar.hist;

cout<<"\n9.Student Geography Marks ::"<<mar.geog;

cout<<"\n10.Student Maths Marks ::"<<mar.math;

cout<<"\n11.Student Economics ::"<<mar.eco;

cout<<"\nTotal Marks ::";

mar.total=mar.hindi+mar.eng+mar.phy+mar.chem

+mar.bio+mar.hist+mar.geog+mar.math+mar.eco;

cout<<mar.total;

mar.com=0,mar.comp=0,mar.hist=0,mar.art=0,mar.acc=0;

do

{

cout<<"\n\nEnter the number written in front feild ::";

cin>>numb;

if(numb==1)

{

cout<<endl<<"Sorry It can't be changed ::";

}

else if(numb==2)

{

cout<<endl<<"Sorry It can't be changed ::";

}

else if(numb==3)

{

cout<<endl<<"New Hindi Marks ::";

cin>>mar.hindi;

}

else if(numb==4)

{

cout<<endl<<"New English Marks ::";

cin>>mar.eng;

}

else if(numb==5)

{

cout<<endl<<"New Physics Marks ::";

cin>>mar.phy;

}

else if(numb==6)

{

cout<<endl<<"New Chemistry Marks ::";

cin>>mar.chem;

}

else if(numb==7)

{

cout<<endl<<"New Biology Marks ::";

cin>>mar.bio;

}

else if(numb==8)

{

cout<<endl<<"New History Marks ::";

cin>>mar.hist;

}

else if(numb==9)

{

cout<<endl<<"New Geography Marks ::";

cin>>mar.geog;

}

else if(numb==10)

{

cout<<endl<<"New Maths Marks ::";

cin>>mar.math;

}

else if(numb==11)

{

cout<<endl<<"New Economic Marks ::";

cin>>mar.eco;

}

else

{

cout<<"\n\nWrong Number choice";

}

cout<<"\n\nWant to change any other(y|n)::";

ans=getche();;

}while(ans=='y');

temp.write((char\*)&mar,sizeof(exam::mark));

}

else

{

temp.write((char\*)&mar,sizeof(exam::mark));

}

}

file.close();

temp.close();

}

else

{

cout<<"\n\nWrong Choice";

}

}

else if(choice>=11 && choice<=12)

{

cout<<"\nSCIENCE SECTION";

cout<<"\nECONOMICS SECTION";

cout<<"\n\n Enter your choice";

cin>>last;

if(last==1)

{

cout<<"\nEnter Student Id ::";

cin>>id;

while(file.read((char\*)&mar,sizeof(exam::mark)))

{

if(id==mar.mark\_id)

{

flag=1;

cout<<"\n1.Student ID ::"<<mar.mark\_id;

cout<<"\n2.Student Class ::"<<mar.clas;

cout<<"\n3.Student Hindi Marks ::"<<mar.hindi;

cout<<"\n4.Student English Marks ::"<<mar.eng;

cout<<"\n5.Student Physics Marks ::"<<mar.phy;

cout<<"\n6.Student Chemistry Marks ::"<<mar.chem;

cout<<"\n7.Student Biology Marks ::"<<mar.bio;

cout<<"\n8.Student Computer Marks ::"<<mar.comp;

cout<<"\n9.Student Maths Marks ::"<<mar.math;

cout<<"\nTotal Marks ::";

mar.total=mar.hindi+mar.eng+mar.phy+mar.chem+

mar.bio+mar.comp+mar.math;

cout<<mar.total;

mar.geog=0,mar.social=0,mar.com=0,mar.science=0,

mar.eco=0,mar.hist=0,mar.art=0,mar.acc=0;

do

{

cout<<"\n\nEnter the number written in front feild ::";

cin>>numb;

if(numb==1)

{

cout<<endl<<"Sorry It can't be changed ::";

}

else if(numb==2)

{

cout<<endl<<"Sorry It can't be changed ::";

}

else if(numb==3)

{

cout<<endl<<"New Hindi Marks ::";

cin>>mar.hindi;

}

else if(numb==4)

{

cout<<endl<<"New English Marks ::";

cin>>mar.eng;

}

else if(numb==5)

{

cout<<endl<<"New Physics Marks ::";

cin>>mar.phy;

}

else if(numb==6)

{

cout<<endl<<"New Chemistry Marks ::";

cin>>mar.chem;

}

else if(numb==7)

{

cout<<endl<<"New Biology Marks ::";

cin>>mar.bio;

}

else if(numb==8)

{

cout<<endl<<"New Computer Marks ::";

cin>>mar.comp;

}

else if(numb==9)

{

cout<<endl<<"New Maths Marks ::";

cin>>mar.math;

}

else

{

cout<<"\n\nWrong Number choice";

}

cout<<"\n\nWant to change any other(y|n)::";

ans=getche();;

}while(ans=='y');

temp.write((char\*)&mar,sizeof(exam::mark));

}

else

{

temp.write((char\*)&mar,sizeof(exam::mark));

}

}

file.close();

temp.close();

}

else if(last==2)

{

cout<<"\nEnter Student Id ::";

cin>>id;

while(file.read((char\*)&mar,sizeof(exam::mark)))

{

if(id==mar.mark\_id)

{

flag=1;

cout<<"\n1.Student ID ::"<<mar.mark\_id;

cout<<"\n2.Student Class ::"<<mar.clas;

cout<<"\n3.Student Hindi Marks ::"<<mar.hindi;

cout<<"\n4.Student English Marks ::"<<mar.eng;

cout<<"\n5.Student Commerce Marks ::"<<mar.com;

cout<<"\n6.Student Economics Marks ::"<<mar.eco;

cout<<"\n7.Student Art Marks ::"<<mar.art;

cout<<"\n8.Student Account Marks ::"<<mar.acc;

cout<<"\n9.Student Computer Marks ::"<<mar.comp;

cout<<"\n10.Student Maths Marks ::"<<mar.math;

cout<<"\nTotal Marks ::";

mar.total=mar.hindi+mar.eng+mar.com+mar.eco+

mar.acc+mar.comp+mar.math+mar.art;

cout<<mar.total;

mar.geog=0,mar.social=0,mar.bio=0,mar.chem=0,mar.phy=0

,mar.science=0,mar.hist=0;

do

{

cout<<"\n\nEnter the number written in front feild ::";

cin>>numb;

if(numb==1)

{

cout<<endl<<"Sorry It can't be changed ::";

}

else if(numb==2)

{

cout<<endl<<"Sorry It can't be changed ::";

}

else if(numb==3)

{

cout<<endl<<"New Hindi Marks ::";

cin>>mar.hindi;

}

else if(numb==4)

{

cout<<endl<<"New English Marks ::";

cin>>mar.eng;

}

else if(numb==5)

{

cout<<endl<<"New Commerce Marks ::";

cin>>mar.com;

}

else if(numb==6)

{

cout<<endl<<"New Economics Marks ::";

cin>>mar.eco;

}

else if(numb==7)

{

cout<<endl<<"New Art Marks ::";

cin>>mar.art;

}

else if(numb==8)

{

cout<<endl<<"New Account Marks ::";

cin>>mar.acc;

}

else if(numb==9)

{

cout<<endl<<"New Computer Marks ::";

cin>>mar.comp;

}

else if(numb==10)

{

cout<<endl<<"New Maths Marks ::";

cin>>mar.math;

}

else

{

cout<<"\n\nWrong Number choice";

}

cout<<"\n\nWant to change any other(y|n)::";

ans=getche();;

}while(ans=='y');

temp.write((char\*)&mar,sizeof(exam::mark));

}

else

{

temp.write((char\*)&mar,sizeof(exam::mark));

}

}

file.close();

temp.close();

}

else

{

cout<<"\n\nWrong Choice";

}

}

else

{

cout<<"\n\nWrong Choice";

}

if(flag==0)

{

cout<<"\n\n Record not found";

}

else

{

file.open(filename,ios::out | ios::binary);

temp.open("temp.dat",ios::in | ios::binary);

while(temp.read((char\*)&mar,sizeof(exam::mark)))

{

file.write((char\*)&mar,sizeof(exam::mark));

}

temp.close();

file.close();

remove("temp.dat");

}

}

getch();

break;

case 3:

clrscr();

cout<<"\nNote::-please write same name otherwise data will be lost\n";

cout<<"\nName of file should be year with class number eg(c2010\_1.dat) for your";

cout<<"\nconvinience";

cout<<"\n\n\nEnter the name of the file::";

gets(filename);

file.open(filename,ios::nocreate | ios::in | ios::binary);

temp.open("temp.dat",ios::out | ios::binary);

file.seekg(0,ios::beg);

if(!file)

{

cout<<"\nSorry Wrong File name is given";

}

else

{

cout<<"\nEnter Student Id ::";

cin>>id;

while(file.read((char\*)&mar,sizeof(exam::mark)))

{

if(id==mar.mark\_id)

{

flag=1;

cout<<"\n\n Record deleted";

}

else

{

temp.write((char\*)&mar,sizeof(exam::mark));

}

}

file.close();

temp.close();

if(flag==0)

{

cout<<"\n\nRecord not found";

}

else

{

file.open(filename,ios::out | ios::binary);

temp.open("temp.dat",ios::in | ios::binary);

while(temp.read((char\*)&mar,sizeof(exam::mark)))

{

file.write((char\*)&mar,sizeof(exam::mark));

}

temp.close();

file.close();

remove("temp.dat");

}

}

getch();

break;

case 4:

clrscr();

cout<<"\nNote::-please write same name otherwise data will be lost\n";

cout<<"\nName of file should be year with class number eg(c2010\_1.dat) for your";

cout<<"\nconvinience";

cout<<"\n\n\nEnter the name of the file::";

gets(filename);

file.open(filename,ios::nocreate | ios::in | ios::binary);

file.seekg(0,ios::beg);

if(!file)

{

cout<<"\nSorry Wrong File name is given";

}

else

{

cout<<"\nEnter Student Id ::";

cin>>id;

while(file.read((char\*)&mar,sizeof(exam::mark)))

{

if(id==mar.mark\_id)

{

flag=1;

cout<<"\nStudent ID ::"<<mar.mark\_id;

cout<<"\nStudent Class ::"<<mar.clas;

cout<<"\nStudent Hindi Marks ::"<<mar.hindi;

cout<<"\nStudent English Marks ::"<<mar.eng;

cout<<"\nStudent Science Marks ::"<<mar.science;

cout<<"\nStudent Physics Marks ::"<<mar.phy;

cout<<"\nStudent Chemistry Marks ::"<<mar.chem;

cout<<"\nStudent Biology Marks ::"<<mar.bio;

cout<<"\nStudent Social Marks ::"<<mar.social;

cout<<"\nStudent History Marks ::"<<mar.hist;

cout<<"\nStudent Geography Marks ::"<<mar.geog;

cout<<"\nStudent Commerce Marks ::"<<mar.com;

cout<<"\nStudent Economic Marks ::"<<mar.eco;

cout<<"\nStudent Account Marks ::"<<mar.acc;

cout<<"\nStudent Art Marks ::"<<mar.art;

cout<<"\nStudent Computer Marks ::"<<mar.comp;

cout<<"\nStudent Maths Marks ::"<<mar.math;

cout<<"\nTotal Marks ::"<<mar.total;

}

}

file.close();

temp.close();

if(flag==0)

{

cout<<"\n\nRecord not found";

}

}

getch();

break;

case 5:

clrscr();

cout<<"\nNote::-please write same name otherwise data will be lost\n";

cout<<"\nName of file should be year with class number eg(c2010\_1.dat) for your";

cout<<"\nconvinience";

cout<<"\n\n\nEnter the name of the file::";

gets(filename);

file.open(filename,ios::nocreate | ios::in | ios::binary);

file.seekg(0,ios::beg);

if(!file)

{

cout<<"\nSorry Wrong File name is given";

}

else

{

while(file.read((char\*)&mar,sizeof(exam::mark)))

{

cout<<"\nStudent ID ::"<<mar.mark\_id;

cout<<"\nStudent Class ::"<<mar.clas;

cout<<"\nStudent Hindi Marks ::"<<mar.hindi;

cout<<"\nStudent English Marks ::"<<mar.eng;

cout<<"\nStudent Science Marks ::"<<mar.science;

cout<<"\nStudent Physics Marks ::"<<mar.phy;

cout<<"\nStudent Chemistry Marks ::"<<mar.chem;

cout<<"\nStudent Biology Marks ::"<<mar.bio;

cout<<"\nStudent Social Marks ::"<<mar.social;

cout<<"\nStudent History Marks ::"<<mar.hist;

cout<<"\nStudent Geography Marks ::"<<mar.geog;

cout<<"\nStudent Commerce Marks ::"<<mar.com;

cout<<"\nStudent Economic Marks ::"<<mar.eco;

cout<<"\nStudent Account Marks ::"<<mar.acc;

cout<<"\nStudent Art Marks ::"<<mar.art;

cout<<"\nStudent Computer Marks ::"<<mar.comp;

cout<<"\nStudent Maths Marks ::"<<mar.math;

cout<<"\nTotal Marks ::"<<mar.total;

cout<<"\n=====================================================================";

getch();

}

file.close();

}

break;

case 6:

menu mn;

mn.start();

break;

default:

cout<<"\n\n Wrong Choice";

break;

}

}

}

//------------------------------------------------MISCELLANEOUS CLASS--------

//====================TECH ADD===============================================

void miscell::teacher\_member()

{

teacher th;

while(1)

{

clrscr();

char ans='n';

int numb=0,id=0,flag=0,last=0;

fstream file,file1,temp,temp1;

int ch;

cout<<"1.Teacher Add New"<<endl<<"2.Teacher Update"<<endl<<"3.Teacher Delete"

<<endl<<"4.Teacher View"<<endl<<"5.Sub Main menu";

cout<<"\n\nEnter your choice ::";

cin>>ch;

switch(ch)

{

case 1:

clrscr();

file.open("teachers.dat",ios::in | ios::binary);

file.seekg(0,ios::beg);

while(file.read((char\*)&th,sizeof(teacher)))

{

last=1;

id=th.teach\_id;

}

if(last==0)

{

id=1;

}

else

{

id=id+1;

}

th.teach\_id=id;

file.close();

file1.open("teachers.dat", ios::app | ios::binary);

cout<<"Teacher id ::";

cout<<th.teach\_id;

cout<<endl;

cout<<"\nTeacher name ::";

gets(th.name);

cout<<endl;

cout<<"Address ::";

gets(th.add);

cout<<endl;

cout<<"Contact ::";

gets(th.phone);

cout<<endl;

cout<<"Qualification ::";

gets(th.qualific);

cout<<endl;

cout<<"Specialization Subject ::";

gets(th.subject);

cout<<endl;

cout<<"Subjects taught ::";

gets(th.taught);

cout<<endl;

cout<<"Class teacher(applicable)::";

gets(th.clas\_teach);

cout<<endl;

cout<<"Activity incharge ::";

gets(th.activity);

file1.write((char\*)&th,sizeof(teacher));

file1.close();

break;

case 2:

clrscr();

file.open("teachers.dat", ios::in | ios::binary);

file1.open("temp.dat",ios::out | ios::binary);

file.seekg(0,ios::beg);

cout<<"Enter the teacher ::";

cin>>id;

while(file.read((char\*)&th,sizeof(teacher)))

{

if(id==th.teach\_id)

{

flag=1;

cout<<endl<<"1.Teacher Id ::"<<th.teach\_id;

cout<<endl<<"2.Teacher Name ::"<<th.name;

cout<<endl<<"3.Address ::"<<th.add;

cout<<endl<<"4.Contact ::"<<th.phone;

cout<<endl<<"5.Qualification ::"<<th.qualific;

cout<<endl<<"6.Specilization ::"<<th.subject;

cout<<endl<<"7.Subject Taught ::"<<th.taught;

cout<<endl<<"8.Class Teacher ::"<<th.clas\_teach;

cout<<endl<<"9.Activity Incharge ::"<<th.activity;

do

{

cout<<"\n\nEnter the number written in front feild ::";

fflush(stdin);

cin>>numb;

if(numb==1)

{

cout<<endl<<"Sorry It can't be changed ::";

}

else if(numb==2)

{

cout<<endl<<"New Teacher Name ::";

gets(th.name);

}

else if(numb==3)

{

cout<<endl<<"New Address ::";

gets(th.add);

}

else if(numb==4)

{

cout<<endl<<"New Contact ::";

gets(th.phone);

}

else if(numb==5)

{

cout<<endl<<"New Qualification ::";

gets(th.qualific);

}

else if(numb==6)

{

cout<<endl<<"New Specilization ::";

gets(th.subject);

}

else if(numb==7)

{

cout<<endl<<"New Subject Taught ::";

gets(th.taught);

}

else if(numb==8)

{

cout<<endl<<"New Class Teacher ::";

gets(th.clas\_teach);

}

else if(numb==9)

{

cout<<endl<<"New Activity Incharge ::";

gets(th.activity);

}

else

{

cout<<"\n\nWrong Number choice";

}

cout<<"\n\nWant to change any other(y|n)::";

ans=getche();

}while(ans=='y');

file1.write((char\*)&th,sizeof(teacher));

}

else

{

file1.write((char\*)&th,sizeof(teacher));

}

}

file.close();

file1.close();

if(flag==0)

{

cout<<"Record not found";

getch();

}

else

{

file.open("temp.dat",ios::in | ios::binary);

file1.open("teachers.dat",ios::out | ios::binary);

file.seekg(0,ios::beg);

while(file.read((char\*)&th,sizeof(teacher)))

{

file1.write((char\*)&th,sizeof(teacher));

}

file.close();

file1.close();

remove("temp.dat");

}

break;

case 3:

clrscr();

file.open("teachers.dat", ios::in | ios::binary);

file1.open("temp.dat",ios::out | ios::binary);

file.seekg(0,ios::beg);

cout<<"Enter the teacher id ::";

cin>>id;

while(file.read((char\*)&th,sizeof(teacher)))

{

if(id==th.teach\_id)

{

flag=1;

cout<<"\n\n Record deleted";

}

else

{

file1.write((char\*)&th,sizeof(teacher));

}

}

file.close();

file1.close();

if(flag==0)

{

cout<<"Record not found";

getch();

}

else

{

file.open("temp.dat",ios::in | ios::binary);

file1.open("teachers.dat",ios::out | ios::binary);

file.seekg(0,ios::beg);

while(file.read((char\*)&th,sizeof(teacher)))

{

file1.write((char\*)&th,sizeof(teacher));

}

file.close();

file1.close();

remove("temp.dat");

}

getch();

break;

case 4:

clrscr();

file.open("teachers.dat", ios::in | ios::binary);

file.seekg(0,ios::beg);

cout<<"Enter the teacher ::";

cin>>id;

while(file.read((char\*)&th,sizeof(teacher)))

{

if(id==th.teach\_id)

{

flag=1;

cout<<endl<<"Teacher Id ::"<<th.teach\_id;

cout<<endl<<"Teacher Name ::"<<th.name;

cout<<endl<<"Address ::"<<th.add;

cout<<endl<<"Contact ::"<<th.phone;

cout<<endl<<"Qualification ::"<<th.qualific;

cout<<endl<<"Specilization ::"<<th.subject;

cout<<endl<<"Subject Taught ::"<<th.taught;

cout<<endl<<"Class Teacher ::"<<th.clas\_teach;

cout<<endl<<"Activity Incharge ::"<<th.activity;

}

}

file.close();

if(flag==0)

{

cout<<"Record not found";

}

getch();

break;

case 5:

this->misc\_submenu();

break;

default:

cout<<"\n\n\n Wrong choice";

getch();

break;

}

}

}

//====================SPORT ADD==============================================

void miscell::sport\_member()

{

sport sp;

while(1)

{

clrscr();

char ans='n';

int numb=0,id=0,flag=0,last=0;

fstream file,file1,temp,temp1;

int ch;

cout<<"1.Sport Add New"<<endl<<"2.Sport Update"<<endl<<"3.Sport Delete"

<<endl<<"4.Sport View"<<endl<<"5.Sub Main menu";

cout<<"\n\nEnter your choice ::";

cin>>ch;

switch(ch)

{

case 1:

clrscr();

file.open("sports.dat",ios::in | ios::binary);

file.seekg(0,ios::beg);

while(file.read((char\*)&sp,sizeof(sport)))

{

last=1;

id=sp.sport\_id;

}

if(last==0)

{

id=1;

}

else

{

id=id+1;

}

sp.sport\_id=id;

file.close();

file1.open("sports.dat", ios::app | ios::binary);

cout<<"Sport id ::";

cout<<sp.sport\_id;

cout<<endl;

cout<<"\nGame name ::";

gets(sp.game);

cout<<endl;

cout<<"Head Coach ::";

gets(sp.head);

cout<<endl;

cout<<"Sub Head Coach ::";

gets(sp.sub\_head);

cout<<endl;

cout<<"Captain ::";

gets(sp.captain);

cout<<endl;

cout<<"Vice Captain ::";

gets(sp.vice);

cout<<endl;

cout<<"Team member ::";

gets(sp.other);

cout<<endl;

cout<<"Level-state,school,etc::";

gets(sp.level);

cout<<endl;

cout<<"Awards ::";

gets(sp.award);

cout<<endl;

cout<<"Position ::";

gets(sp.position);

file1.write((char\*)&sp,sizeof(sport));

file1.close();

break;

case 2:

clrscr();

file.open("sports.dat", ios::in | ios::binary);

file1.open("temp.dat",ios::out | ios::binary);

file.seekg(0,ios::beg);

cout<<"Enter the Sport Id ::";

cin>>id;

while(file.read((char\*)&sp,sizeof(sport)))

{

if(id==sp.sport\_id)

{

flag=1;

cout<<endl<<"1.Sport Id ::"<<sp.sport\_id;

cout<<endl<<"2.Game Name ::"<<sp.game;

cout<<endl<<"3.Head Coach ::"<<sp.head;

cout<<endl<<"4.Sub Coach ::"<<sp.sub\_head;

cout<<endl<<"5.Captain ::"<<sp.captain;

cout<<endl<<"6.Vice Captain ::"<<sp.vice;

cout<<endl<<"7.Team Members ::"<<sp.other;

cout<<endl<<"8.Level ::"<<sp.level;

cout<<endl<<"9.Awards ::"<<sp.award;

cout<<endl<<"10.Position ::"<<sp.position;

do

{

cout<<"\n\nEnter the number written in front feild ::";

cin>>numb;

if(numb==1)

{

cout<<endl<<"Sorry It can't be changed ::";

}

else if(numb==2)

{

cout<<endl<<"New Game Name ::";

gets(sp.game);

}

else if(numb==3)

{

cout<<endl<<"New Head Coach ::";

gets(sp.head);

}

else if(numb==4)

{

cout<<endl<<"New Sub Coach ::";

gets(sp.sub\_head);

}

else if(numb==5)

{

cout<<endl<<"New Captain ::";

gets(sp.captain);

}

else if(numb==6)

{

cout<<endl<<"New Vice Captain ::";

gets(sp.vice);

}

else if(numb==7)

{

cout<<endl<<"New Team Members ::";

gets(sp.other);

}

else if(numb==8)

{

cout<<endl<<"New Level ::";

gets(sp.level);

}

else if(numb==9)

{

cout<<endl<<"New Awards ::";

gets(sp.award);

}

else if(numb==10)

{

cout<<endl<<"New Position ::";

gets(sp.position);

}

else

{

cout<<"\n\nWrong Number choice";

}

cout<<"\n\nWant to change any other(y|n)::";

ans=getche();

}while(ans=='y');

file1.write((char\*)&sp,sizeof(sport));

}

else

{

file1.write((char\*)&sp,sizeof(sport));

}

}

file.close();

file1.close();

if(flag==0)

{

cout<<"Record not found";

getch();

}

else

{

file.open("temp.dat",ios::in | ios::binary);

file1.open("sports.dat",ios::out | ios::binary);

file.seekg(0,ios::beg);

while(file.read((char\*)&sp,sizeof(sport)))

{

file1.write((char\*)&sp,sizeof(sport));

}

file.close();

file1.close();

remove("temp.dat");

}

break;

case 3:

clrscr();

file.open("sports.dat", ios::in | ios::binary);

file1.open("temp.dat",ios::out | ios::binary);

file.seekg(0,ios::beg);

cout<<"Enter the Sport Id ::";

cin>>id;

while(file.read((char\*)&sp,sizeof(sport)))

{

if(id==sp.sport\_id)

{

flag=1;

cout<<"\n\n Record deleted";

}

else

{

file1.write((char\*)&sp,sizeof(sport));

}

}

file.close();

file1.close();

if(flag==0)

{

cout<<"Record not found";

getch();

}

else

{

file.open("temp.dat",ios::in | ios::binary);

file1.open("sports.dat",ios::out | ios::binary);

file.seekg(0,ios::beg);

while(file.read((char\*)&sp,sizeof(sport)))

{

file1.write((char\*)&sp,sizeof(sport));

}

file.close();

file1.close();

remove("temp.dat");

}

getch();

break;

case 4:

clrscr();

file.open("sports.dat", ios::in | ios::binary);

file.seekg(0,ios::beg);

cout<<"Enter the Sport Id ::";

cin>>id;

while(file.read((char\*)&sp,sizeof(sport)))

{

if(id==sp.sport\_id)

{

flag=1;

cout<<endl<<"Sport Id ::"<<sp.sport\_id;

cout<<endl<<"Game Name ::"<<sp.game;

cout<<endl<<"Head Coach ::"<<sp.head;

cout<<endl<<"Sub Coach ::"<<sp.sub\_head;

cout<<endl<<"Captain ::"<<sp.captain;

cout<<endl<<"Vice Captain ::"<<sp.vice;

cout<<endl<<"Team Members ::"<<sp.other;

cout<<endl<<"Level ::"<<sp.level;

cout<<endl<<"Awards ::"<<sp.award;

cout<<endl<<"Position ::"<<sp.position;

}

}

file.close();

if(flag==0)

{

cout<<"Record not found";

}

getch();

break;

case 5:

this->misc\_submenu();

break;

default:

cout<<"\n\n\n Wrong Choice";

getch();

break;

}

}

}

//====================STAFF ADD==============================================

void miscell::staff\_member()

{

staff st;

while(1)

{

clrscr();

char ans='n';

int numb=0,id=0,flag=0,last=0;

fstream file,file1,temp,temp1;

int ch;

cout<<"1.Staff Add New"<<endl<<"2.Staff Update"<<endl<<"3.Staff Delete"

<<endl<<"4.Staff View"<<endl<<"5.Sub Main menu";

cout<<"\n\nEnter your choice ::";

cin>>ch;

switch(ch)

{

case 1:

clrscr();

file.open("staffs.dat",ios::in | ios::binary);

file.seekg(0,ios::beg);

while(file.read((char\*)&st,sizeof(staff)))

{

last=1;

id=st.staff\_id;

}

if(last==0)

{

id=1;

}

else

{

id=id+1;

}

st.staff\_id=id;

file.close();

file1.open("staffs.dat", ios::app | ios::binary);

cout<<"Staff Id ::";

cout<<st.staff\_id<<endl;

cout<<"\nStaff name ::";

gets(st.name);

cout<<endl;

cout<<"Address ::";

gets(st.add);

cout<<endl;

cout<<"Contact ::";

gets(st.phone);

cout<<endl;

cout<<"Retired(yes/no) ::";

gets(st.retired);

cout<<endl;

cout<<"Job ::";

gets(st.job);

cout<<endl;

cout<<"Post ::";

gets(st.post);

file1.write((char\*)&st,sizeof(staff));

file1.close();

break;

case 2:

clrscr();

file.open("staffs.dat", ios::in | ios::binary);

file1.open("temp.dat",ios::out | ios::binary);

file.seekg(0,ios::beg);

cout<<"Enter the Staff id ::";

cin>>id;

while(file.read((char\*)&st,sizeof(staff)))

{

if(id==st.staff\_id)

{

flag=1;

cout<<endl<<"1.Staff Id ::"<<st.staff\_id;

cout<<endl<<"2.Staff Name ::"<<st.name;

cout<<endl<<"3.Address ::"<<st.add;

cout<<endl<<"4.Contact ::"<<st.phone;

cout<<endl<<"5.Retired ::"<<st.retired;

cout<<endl<<"6.Job ::"<<st.job;

cout<<endl<<"7.Post ::"<<st.post;

do

{

cout<<"\n\nEnter the number written in front feild ::";

cin>>numb;

if(numb==1)

{

cout<<endl<<"Sorry It can't be changed ::";

}

else if(numb==2)

{

cout<<endl<<"New Staff Name ::";

gets(st.name);

}

else if(numb==3)

{

cout<<endl<<"New Address ::";

gets(st.add);

}

else if(numb==4)

{

cout<<endl<<"New Contact ::";

gets(st.phone);

}

else if(numb==5)

{

cout<<endl<<"New Retired ::";

gets(st.retired);

}

else if(numb==6)

{

cout<<endl<<"New Job ::";

gets(st.job);

}

else if(numb==7)

{

cout<<endl<<"New Post ::";

gets(st.post);

}

else

{

cout<<"\n\nWrong Number choice";

}

cout<<"\n\nWant to change any other(y|n)::";

ans=getche();

}while(ans=='y');

file1.write((char\*)&st,sizeof(staff));

}

else

{

file1.write((char\*)&st,sizeof(staff));

}

}

file.close();

file1.close();

if(flag==0)

{

cout<<"Record not found";

getch();

}

else

{

file.open("temp.dat",ios::in | ios::binary);

file1.open("staffs.dat",ios::out | ios::binary);

file.seekg(0,ios::beg);

while(file.read((char\*)&st,sizeof(staff)))

{

file1.write((char\*)&st,sizeof(staff));

}

file.close();

file1.close();

remove("temp.dat");

}

break;

case 3:

clrscr();

file.open("staffs.dat", ios::in | ios::binary);

file1.open("temp.dat",ios::out | ios::binary);

file.seekg(0,ios::beg);

cout<<"Enter the Staff id ::";

cin>>id;

while(file.read((char\*)&st,sizeof(staff)))

{

if(id==st.staff\_id)

{

flag=1;

cout<<"\n\n Record deleted";

}

else

{

file1.write((char\*)&st,sizeof(staff));

}

}

file.close();

file1.close();

if(flag==0)

{

cout<<"Record not found";

}

else

{

file.open("temp.dat",ios::in | ios::binary);

file1.open("staffs.dat",ios::out | ios::binary);

file.seekg(0,ios::beg);

while(file.read((char\*)&st,sizeof(staff)))

{

file1.write((char\*)&st,sizeof(staff));

}

file.close();

file1.close();

remove("temp.dat");

}

getch();

break;

case 4:

clrscr();

file.open("staffs.dat", ios::in | ios::binary);

file.seekg(0,ios::beg);

cout<<"Enter the Staff id ::";

cin>>id;

while(file.read((char\*)&st,sizeof(staff)))

{

if(id==st.staff\_id)

{

flag=1;

cout<<endl<<"Staff Id ::"<<st.staff\_id;

cout<<endl<<"Staff Name ::"<<st.name;

cout<<endl<<"Address ::"<<st.add;

cout<<endl<<"Contact ::"<<st.phone;

cout<<endl<<"Retired ::"<<st.retired;

cout<<endl<<"Job ::"<<st.job;

cout<<endl<<"Post ::"<<st.post;

}

}

file.close();

if(flag==0)

{

cout<<"Record not found";

}

getch();

break;

case 5:

this->misc\_submenu();

break;

default:

cout<<"\n\n\n Wrong choice";

getch();

break;

}

}

}

//---------------------------------------------ACCOUNT CLASS FUNCTION--------

//====================SALARY=================================================

void account::salary\_account()//miscell mis,account acc)

{

account::salary sal;

while(1)

{

clrscr();

char ans='n',filename[25];

int numb=0,cal=0,id=0,flag=0,choice=0,already=0;

int ch;

fstream file,file1,temp;

cout<<"1.Salary Add New"<<endl<<"2.Salary Update"<<endl<<"3.Salary Delete"

<<endl<<"4.Salary View"<<endl<<"5.Sub Main menu";

cout<<"\n\nEnter your choice ::";

cin>>ch;

switch(ch)

{

case 1:

clrscr();

miscell::teacher th;

miscell::staff st;

cout<<"1.This month press 1 ::"<<endl;

cout<<"2.Previous month press 2::"<<endl<<endl;

cout<<"Enter your choice::";

cin>>choice;

if(choice==1)

{

cout<<"\nNote::-please write same name otherwise data will be lost\n";

cout<<"\nName of file should be year with month number eg(f2010\_1.dat) for your";

cout<<"\nconvinience";

cout<<"\n\n\nEnter the name of the file::";

gets(filename);

file.open(filename,ios::app | ios::binary);

cout<<"\nEnter 1 for teacher or 2 for staff salary ::";

cin>>numb;

if(numb==1)

{

strcpy(sal.teach,"teacher");

cout<<"\nEnter Id ::";

cin>>id;

file1.open("teachers.dat",ios::in | ios::binary);

temp.open(filename,ios::in |ios::binary);

while(file1.read((char\*)&th,sizeof(miscell::teacher)))

{

if(id==th.teach\_id)

{

flag=1;

}

}

file1.close();

if(flag==1)

{

while(temp.read((char\*)&sal,sizeof(account::salary)))

{

if(id==sal.sal\_id && strcmp(sal.teach,"teacher")==0)

{

already=1;

}

}

temp.close();

if(already==0)

{

sal.sal\_id=id;

cout<<"\nEnter Basic Salary ::";

cin>>sal.basic;

cout<<"\nEnter DA for Salary ::";

cin>>sal.da;

cout<<"\nEnter House rent only if premises not provided::";

cin>>sal.hr;

cout<<"\nEnter Medial service MC ::";

cin>>sal.mc;

cout<<"\nEnter PA ::";

cin>>sal.pa;

file.write((char\*)&sal,sizeof(account::salary));

cal=sal.basic+sal.da+sal.hr+sal.mc-sal.pa;

cout<<"\nTotal Salary ::"<<cal;

}

else

{

cout<<"\n\n Already exist";

}

}

else

{

cout<<"\n\n\n No Such Id";

}

}

else if(numb==2)

{

strcpy(sal.teach,"staff");

cout<<"\nEnter Id ::";

cin>>id;

file1.open("staffs.dat",ios::in | ios::binary);

while(file1.read((char\*)&st,sizeof(miscell::staff)))

{

if(id==st.staff\_id)

{

flag=1;

}

}

file1.close();

if(flag==1)

{

while(temp.read((char\*)&sal,sizeof(account::salary)))

{

if(id==sal.sal\_id && strcmp(sal.teach,"staff")==0)

{

already=1;

}

}

temp.close();

if(already==0)

{

sal.sal\_id=id;

cout<<"\nEnter Basic Salary ::";

cin>>sal.basic;

cout<<"\nEnter DA for Salary ::";

cin>>sal.da;

cout<<"\nEnter House rent only if premises not provided::";

cin>>sal.hr;

cout<<"\nEnter Medial service MC ::";

cin>>sal.mc;

cout<<"\nEnter PA ::";

cin>>sal.pa;

file.write((char\*)&sal,sizeof(account::salary));

cal=sal.basic+sal.da+sal.hr+sal.mc-sal.pa;

cout<<"\nTotal Salary ::"<<cal;

}

else

{

cout<<"\n\n Already exist";

}

}

else

{

cout<<"\n\n\n No Such Id";

}

}

else

{

cout<<"\n\n\n Wrong Choice";

}

file.close();

}

else if(choice==2)

{

cout<<"\nName of file should be year with month number eg(f2010\_1.dat) for your";

cout<<"\nconvinience";

cout<<"\n\n\nEnter the name of the file::";

gets(filename);

file.open(filename,ios::nocreate | ios::app | ios::binary );

if(!file)

{

cout<<"\n\n\Sorry Wrong name mentioned not available";

}

else

{

cout<<"\nEnter 1 for teacher or 2 for staff salary ::";

cin>>numb;

if(numb==1)

{

strcpy(sal.teach,"teacher");

cout<<"\nEnter Id ::";

cin>>id;

file1.open("teachers.dat",ios::in | ios::binary);

while(file1.read((char\*)&th,sizeof(miscell::teacher)))

{

if(id==th.teach\_id)

{

flag=1;

}

}

file1.close();

if(flag==1)

{

while(temp.read((char\*)&sal,sizeof(account::salary)))

{

if(id==sal.sal\_id && strcmp(sal.teach,"teacher")==0)

{

already=1;

}

}

temp.close();

if(already==0)

{

sal.sal\_id=id;

cout<<"\nEnter Basic Salary ::";

cin>>sal.basic;

cout<<"\nEnter DA for Salary ::";

cin>>sal.da;

cout<<"\nEnter House rent only if premises not provided::";

cin>>sal.hr;

cout<<"\nEnter Medial service MC ::";

cin>>sal.mc;

cout<<"\nEnter PA ::";

cin>>sal.pa;

file.write((char\*)&sal,sizeof(account::salary));

}

else

{

cout<<"\n\n Already exist";

}

}

else

{

cout<<"\n\n\n No Such Id";

}

}

else if(numb==2)

{

strcpy(sal.teach,"staff");

cout<<"\nEnter Id ::";

cin>>id;

file1.open("staffs.dat",ios::in | ios::binary);

while(file1.read((char\*)&st,sizeof(miscell::staff)))

{

if(id==st.staff\_id)

{

flag=1;

}

}

file1.close();

if(flag==1)

{

while(temp.read((char\*)&sal,sizeof(account::salary)))

{

if(id==sal.sal\_id && strcmp(sal.teach,"staff")==0)

{

already=1;

}

}

temp.close();

if(already==0)

{

sal.sal\_id=id;

cout<<"\nEnter Basic Salary ::";

cin>>sal.basic;

cout<<"\nEnter DA for Salary ::";

cin>>sal.da;

cout<<"\nEnter House rent only if premises not provided::";

cin>>sal.hr;

cout<<"\nEnter Medial service MC ::";

cin>>sal.mc;

cout<<"\nEnter PA ::";

cin>>sal.pa;

file.write((char\*)&sal,sizeof(account::salary));

}

else

{

cout<<"\n\nAlready exist";

}

}

else

{

cout<<"\n\n\n No Such Id";

}

}

else

{

cout<<"\n\n\n Wrong Choice";

}

file.close();

}

}

else

{

cout<<"\n\n Wrong Choice";

}

getch();

break;

case 2:

clrscr();

cout<<"\nName of file should be year with month number eg(f2010\_1.dat) for your";

cout<<"\nconvinience";

cout<<"\n\n\nEnter the name of the file::";

gets(filename);

file.open(filename,ios::nocreate | ios::in | ios::binary);

temp.open("temp.dat",ios::out | ios::binary);

if(!file)

{

cout<<"\n\n\Sorry Wrong name mentioned not available";

}

else

{ cout<<"\nEnter 1 for teacher or 2 for other staff ::";

cin>>numb;

if(numb==1)

{ cout<<"\nEnter the Teacher Id::";

cin>>id;

while(file.read((char\*)&sal,sizeof(account::salary)))

{

if(id==sal.sal\_id && strcmp(sal.teach,"teacher")==0)

{

flag=1;

cout<<"\n\n";

cout<<"\n1.Staff Id ::"<<sal.sal\_id;

cout<<"\n2.Teacher or Other Staff::"<<sal.teach;

cout<<"\n3.Basic Salary ::"<<sal.basic;

cout<<"\n4.DA ::"<<sal.da;

cout<<"\n5.House Rent ::"<<sal.hr;

cout<<"\n6.Medical Checkup ::"<<sal.mc;

cout<<"\n7.PA ::"<<sal.pa;

do

{

cout<<"\n\nEnter the number written in front feild ::";

cin>>numb;

if(numb==1)

{

cout<<endl<<"Sorry It can't be changed ::";

}

else if(numb==2)

{

cout<<endl<<"Sorry It can't be changed ::";

}

else if(numb==3)

{

cout<<endl<<"New Basic Salary ::";

cin>>sal.basic;

}

else if(numb==4)

{

cout<<endl<<"New DA ::";

cin>>sal.da;

}

else if(numb==5)

{

cout<<endl<<"New House Rent ::";

cin>>sal.hr;

}

else if(numb==6)

{

cout<<endl<<"New Medical Checkup ::";

cin>>sal.mc;

}

else if(numb==7)

{

cout<<endl<<"New PA ::";

cin>>sal.pa;

}

else

{

cout<<"\n\nWrong Number choice";

}

cout<<"\n\nWant to change any other(y|n)::";

ans=getche();

}while(ans=='y');

temp.write((char\*)&sal,sizeof(account::salary));

}

else

{

temp.write((char\*)&sal,sizeof(account::salary));

}

}

}

else if(numb==2)

{

cout<<"\nEnter the Staff Id::";

cin>>id;

while(file.read((char\*)&sal,sizeof(account::salary)))

{

if(id==sal.sal\_id && strcmp(sal.teach,"staff")==0)

{

flag=1;

cout<<"\n\n";

cout<<"\nStaff Id ::"<<sal.sal\_id;

cout<<"\nTeacher or Other Staff::"<<sal.teach;

cout<<"\nBasic Salary ::"<<sal.basic;

cout<<"\nDA ::"<<sal.da;

cout<<"\nHouse Rent ::"<<sal.hr;

cout<<"\nMedical Checkup ::"<<sal.mc;

cout<<"\nPA ::"<<sal.pa;

do

{

cout<<"\n\nEnter the number written in front feild ::";

cin>>numb;

if(numb==1)

{

cout<<endl<<"Sorry It can't be changed ::";

}

else if(numb==2)

{

cout<<endl<<"Sorry It can't be changed ::";

}

else if(numb==3)

{

cout<<endl<<"New Basic Salary ::";

cin>>sal.basic;

}

else if(numb==4)

{

cout<<endl<<"New DA ::";

cin>>sal.da;

}

else if(numb==5)

{

cout<<endl<<"New House Rent ::";

cin>>sal.hr;

}

else if(numb==6)

{

cout<<endl<<"New Medical Checkup ::";

cin>>sal.mc;

}

else if(numb==7)

{

cout<<endl<<"New PA ::";

cin>>sal.pa;

}

else

{

cout<<"\n\nWrong Number choice";

}

cout<<"\n\nWant to change any other(y|n)::";

ans=getche();

}while(ans=='y');

temp.write((char\*)&sal,sizeof(account::salary));

}

else

{

temp.write((char\*)&sal,sizeof(account::salary));

}

}

}

else

{

cout<<"\n\n Wrong Choice";

}

file.close();

temp.close();

if(flag==0)

{

cout<<"\n\nRecord not found";

}

else

{

file.open(filename,ios::out | ios::binary);

temp.open("temp.dat",ios::in | ios::binary);

temp.seekg(0,ios::beg);

while(temp.read((char\*)&sal,sizeof(account::salary)))

{

file.write((char\*)&sal,sizeof(account::salary));

}

file.close();

temp.close();

remove("temp.dat");

}

}

getch();

break;

case 3:

clrscr();

cout<<"\nName of file should be year with month number eg(f2010\_1.dat) for your ";

cout<<"\nconvinience";

cout<<"\n\n\nEnter the name of the file::";

gets(filename);

file.open(filename,ios::nocreate | ios::in | ios::binary);

temp.open("temp.dat",ios::out | ios::binary);

if(!file)

{

cout<<"\n\n\Sorry Wrong name mentioned not available";

}

else

{

cout<<"\nEnter 1 for teacher or 2 for other staff ::";

cin>>numb;

if(numb==1)

{

cout<<"\nEnter the Teacher Id::";

cin>>id;

while(file.read((char\*)&sal,sizeof(account::salary)))

{

if(id==sal.sal\_id && strcmp(sal.teach,"teacher")==0)

{

flag=1;

cout<<"\nRecord deleted";

}

else

{

temp.write((char\*)&sal,sizeof(account::salary));

}

}

}

else if(numb==2)

{

cout<<"\nEnter the Staff Id::";

cin>>id;

while(file.read((char\*)&sal,sizeof(account::salary)))

{

if(id==sal.sal\_id && strcmp(sal.teach,"staff")==0)

{

flag=1;

cout<<"\nRecord deleted";

}

else

{

temp.write((char\*)&sal,sizeof(account::salary));

}

}

}

else

{

cout<<"\n\nWrong choice";

}

file.close();

temp.close();

if(flag==0)

{

cout<<"\n\nRecord not found";

}

else

{

file.open(filename,ios::out | ios::binary);

temp.open("temp.dat",ios::in | ios::binary);

temp.seekg(0,ios::beg);

while(temp.read((char\*)&sal,sizeof(account::salary)))

{

file.write((char\*)&sal,sizeof(account::salary));

}

file.close();

temp.close();

remove("temp.dat");

}

}

getch();

break;

case 4:

clrscr();

cout<<"\nName of file should be year with month number eg(f2010\_1.dat) for your";

cout<<"\nconvinience";

cout<<"\n\n\nEnter the name of the file you want to see::";

gets(filename);

file.open(filename,ios::nocreate | ios::in | ios::binary);

if(!file)

{

cout<<"\n\n\Sorry Wrong name mentioned not available";

}

else

{

cout<<"\nEnter 1 for teacher or 2 for other staff ::";

cin>>numb;

if(numb==1)

{

cout<<"\nEnter the Teacher Id::";

cin>>id;

while(file.read((char\*)&sal,sizeof(account::salary)))

{

if(id==sal.sal\_id && strcmp(sal.teach,"teacher")==0)

{

flag=1;

cout<<"\n\n";

cout<<"\nStaff Id ::"<<sal.sal\_id;

cout<<"\nTeacher or Other Staff::"<<sal.teach;

cout<<"\nBasic Salary ::"<<sal.basic;

cout<<"\nDA ::"<<sal.da;

cout<<"\nHouse Rent ::"<<sal.hr;

cout<<"\nMedical Checkup ::"<<sal.mc;

cout<<"\nPA ::"<<sal.pa;

cal=sal.basic+sal.da+sal.hr+sal.mc-sal.pa;

cout<<"\nTotal Salary ::"<<cal;

}

}

}

else if(numb==2)

{

cout<<"\nEnter the Staff Id::";

cin>>id;

while(file.read((char\*)&sal,sizeof(account::salary)))

{

if(id==sal.sal\_id && strcmp(sal.teach,"staff")==0)

{

flag=1;

cout<<"\n\n";

cout<<"\nStaff Id ::"<<sal.sal\_id;

cout<<"\nTeacher or Other Staff::"<<sal.teach;

cout<<"\nBasic Salary ::"<<sal.basic;

cout<<"\nDA ::"<<sal.da;

cout<<"\nHouse Rent ::"<<sal.hr;

cout<<"\nMedical Checkup ::"<<sal.mc;

cout<<"\nPA ::"<<sal.pa;

cal=sal.basic+sal.da+sal.hr+sal.mc-sal.pa;

cout<<"\nTotal Salary ::"<<cal;

}

}

}

else

{

cout<<"\n\n Wrong Choice";

}

if(flag==0)

{

cout<<"\n\n Record not found";

}

file.close();

}

getch();

break;

case 5:

this->account\_submenu();

break;

default:

cout<<"\n\n\nWrong Choice";

getch();

break;

}

}

}

//====================FEE'S==================================================

void account::fee\_account()

{

account::fee fe;

while(1)

{

clrscr();

char ans='n',filename[25];

int id=0,flag=0,numb=0,last=0,choice=0,cal=0;

fstream file,file1,temp,temp1;

int ch;

cout<<"1.Fee Add New"<<endl<<"2.Fee Update"<<endl<<"3.Fee Delete"

<<endl<<"4.Fee View"<<endl<<"5.Sub Main menu";

cout<<"\n\nEnter your choice ::";

cin>>ch;

switch(ch)

{

case 1:

clrscr();

student::stud st;

cout<<"1.This month press 1 ::"<<endl;

cout<<"2.Previous month press 2::"<<endl<<endl;

cout<<"Enter your choice::";

cin>>choice;

if(choice==1)

{

cout<<"\nNote::-please write same name otherwise data will be lost\n";

cout<<"\nName of file should be year with month number eg(s2010\_1.dat) for your";

cout<<"\nconvinience";

cout<<"\n\n\nEnter the name of the file::";

gets(filename);

cout<<"\nEnter Student Id ::";

cin>>id;

file1.open("admission.dat",ios::in | ios::binary);

while(file1.read((char\*)&st,sizeof(student::stud)))

{

if(id==st.student\_id)

{

flag=1;

}

}

file1.close();

if(flag==1)

{

temp.open(filename,ios::in |ios::binary);

temp.seekg(0,ios::beg);

while(temp.read((char\*)&fe,sizeof(account::fee)))

{ if(id==fe.fee\_id)

{

last=1;

}

}

temp.close();

if(last==0)

{

file.open(filename,ios::app | ios::binary);

fe.fee\_id=id;

cout<<"\nEnter Tution Fees ::";

cin>>fe.tuition;

cout<<"\nEnter Bus Service ::";

cin>>fe.bus;

cout<<"\nEnter Magzine Fees ::";

cin>>fe.magzine;

cout<<"\nEnter Misc Fees ::";

cin>>fe.misc;

cout<<"\nEnter Tour Fees ::";

cin>>fe.tour;

cout<<"\nEnter Sport Fees ::";

cin>>fe.sport;

cal=fe.tuition+fe.bus+fe.magzine+fe.misc+fe.tour+fe.sport;

cout<<"\nTotal Fees ::"<<cal;

file.write((char\*)&fe,sizeof(account::fee));

}

else

{

cout<<"\n\n Already exist";

}

file.close();

}

else

{

cout<<"\n\nNo such id";

}

}

else if(choice==2)

{

cout<<"\nNote::-please write same name otherwise data will be lost\n";

cout<<"\nName of file should be year with month number eg(s2010\_1.dat) for your";

cout<<"\nconvinience";

cout<<"\n\n\nEnter the name of the file::";

gets(filename);

file.open(filename,ios::nocreate | ios::app | ios::binary);

if(!file)

{

cout<<"\n\n\Sorry Wrong name mentioned not available";

}

else

{ cout<<"\nEnter Id ::";

cin>>id;

file1.open("admission.dat",ios::in | ios::binary);

while(file1.read((char\*)&st,sizeof(student::stud)))

{ if(id==st.student\_id)

{

flag=1;

}

}

file1.close();

if(flag==1)

{

temp.open(filename,ios::in |ios::binary);

temp.seekg(0,ios::beg);

while(temp.read((char\*)&fe,sizeof(account::fee)))

{ if(id==fe.fee\_id)

{

last=1;

}

}

temp.close();

if(last==0)

{

fe.fee\_id=id;

cout<<"\nEnter Tution Fees ::";

cin>>fe.tuition;

cout<<"\nEnter Bus Service ::";

cin>>fe.bus;

cout<<"\nEnter Magzine Fees ::";

cin>>fe.magzine;

cout<<"\nEnter Misc Fees ::";

cin>>fe.misc;

cout<<"\nEnter Tour Fees ::";

cin>>fe.tour;

cout<<"\nEnter Sport Fees ::";

cin>>fe.sport;

cal=fe.tuition+fe.bus+fe.magzine+fe.misc+fe.tour+fe.sport;

cout<<"\nTotal Fees ::"<<cal;

file.write((char\*)&fe,sizeof(account::fee));

}

else

{

cout<<"\n\n Already exist";

}

file.close();

}

else

{

cout<<"\n\nNo such id";

}

}

}

else

{

cout<<"\n\n Wrong choice";

}

getch();

break;

case 2:

clrscr();

cout<<"\nName of file should be year with month number eg(s2010\_1.dat) for your";

cout<<"\nconvinience";

cout<<"\n\n\nEnter the name of the file::";

gets(filename);

file.open(filename,ios::nocreate | ios::in | ios::binary);

temp.open("temp.dat",ios::out | ios::binary);

if(!file)

{

cout<<"\n\n\Sorry Wrong name mentioned not available";

}

else

{

cout<<"\nEnter the Student Id ::";

cin>>id;

while(file.read((char\*)&fe,sizeof(account::fee)))

{

if(id==fe.fee\_id)

{

flag=1;

cout<<"\n\n";

cout<<"\n1.Student Id ::"<<fe.fee\_id;

cout<<"\n2.Tution Fees ::"<<fe.tuition;

cout<<"\n3.Bus Service ::"<<fe.bus;

cout<<"\n4.Magzine Fees ::"<<fe.magzine;

cout<<"\n5.Misc Fees ::"<<fe.misc;

cout<<"\n6.Tour Fees ::"<<fe.tour;

cout<<"\n7.Sport Fees ::"<<fe.sport;

cal=fe.tuition+fe.bus+fe.magzine+fe.misc+fe.tour+fe.sport;

cout<<"\nTotal Fees ::"<<cal;

do

{

cout<<"\n\nEnter the number written in front feild ::";

cin>>numb;

if(numb==1)

{

cout<<endl<<"Sorry It can't be changed ::";

}

else if(numb==2)

{

cout<<endl<<"New Tuition Fees ::";

cin>>fe.tuition;

}

else if(numb==3)

{

cout<<endl<<"New Bus Fees ::";

cin>>fe.bus;

}

else if(numb==4)

{

cout<<endl<<"New Magzine Fees ::";

cin>>fe.magzine;

}

else if(numb==5)

{

cout<<endl<<"New Misc Fees ::";

cin>>fe.misc;

}

else if(numb==6)

{

cout<<endl<<"New Tour Fees ::";

cin>>fe.tour;

}

else if(numb==7)

{

cout<<endl<<"New Sport Fees ::";

cin>>fe.sport;

}

else

{

cout<<"\n\nWrong Number choice";

}

cout<<"\n\nWant to change any other(y|n)::";

ans=getche();

}while(ans=='y');

temp.write((char\*)&fe,sizeof(account::fee));

}

else

{

temp.write((char\*)&fe,sizeof(account::fee));

}

}

file.close();

temp.close();

if(flag==0)

{

cout<<"\n\nRecord not found";

}

else

{

file.open(filename,ios::out | ios::binary);

temp.open("temp.dat",ios::in | ios::binary);

temp.seekg(0,ios::beg);

while(temp.read((char\*)&fe,sizeof(account::fee)))

{

file.write((char\*)&fe,sizeof(account::fee));

}

file.close();

temp.close();

remove("temp.dat");

}

}

getch();

break;

case 3:

clrscr();

cout<<"\nName of file should be year with month number eg(s2010\_1.dat) for your";

cout<<"\nconv]inience";

cout<<"\n\n\nEnter the name of the file::";

gets(filename);

file.open(filename,ios::nocreate | ios::in | ios::binary);

temp.open("temp.dat",ios::out | ios::binary);

if(!file)

{

cout<<"\n\n\Sorry Wrong name mentioned not available";

}

else

{

cout<<"\nEnter the Student Id ::";

cin>>id;

while(file.read((char\*)&fe,sizeof(account::fee)))

{

if(id==fe.fee\_id)

{

flag=1;

cout<<"\nRecord deleted";

}

else

{

temp.write((char\*)&fe,sizeof(account::fee));

}

}

file.close();

temp.close();

if(flag==0)

{

cout<<"\n\nRecord not found";

}

else

{

file.open(filename,ios::out | ios::binary);

temp.open("temp.dat",ios::in | ios::binary);

temp.seekg(0,ios::beg);

while(temp.read((char\*)&fe,sizeof(account::fee)))

{

file.write((char\*)&fe,sizeof(account::fee));

}

file.close();

temp.close();

remove("temp.dat");

}

}

getch();

break;

case 4:

clrscr();

cout<<"\nName of file should be year with month number eg(s2010\_1.dat) for your";

cout<<"\nconvinience";

cout<<"\n\n\nEnter the name of the file ypu want to see::";

gets(filename);

file.open(filename,ios::nocreate | ios::in | ios::binary);

if(!file)

{

cout<<"\n\n\Sorry Wrong name mentioned not available";

}

else

{

cout<<"\nEnter the Student id ::";

cin>>id;

while(file.read((char\*)&fe,sizeof(account::fee)))

{

if(id==fe.fee\_id)

{

flag=1;

cout<<"\n\n";

cout<<"\nStudent Id ::"<<fe.fee\_id;

cout<<"\nTution Fees ::"<<fe.tuition;

cout<<"\nBus Service ::"<<fe.bus;

cout<<"\nMagzine Fees ::"<<fe.magzine;

cout<<"\nMisc Fees ::"<<fe.misc;

cout<<"\nTour Fees ::"<<fe.tour;

cout<<"\nSport Fees ::"<<fe.sport;

cal=fe.tuition+fe.bus+fe.magzine+fe.misc+fe.tour+fe.sport;

cout<<"\nTotal Fees ::"<<cal;

}

}

if(flag==0)

{

cout<<"\n\n Record not found";

}

file.close();

}

getch();

break;

case 5:

this->account\_submenu();

break;

default:

cout<<"\n\n\nWrong Choice";

getch();

break;

}

}

}

//====================FINE===========================================

void account::fine\_account()

{

account::fine fi;

while(1)

{

clrscr();

char ans='n';

int numb=0,id=0,flag=0;

fstream file,file1,temp;

int ch;

cout<<"1.Fine Add New"<<endl<<"2.Fine Update"<<endl<<"3.Fine Delete"

<<endl<<"4.Fine View"<<endl<<"5.Sub Main menu";

cout<<"\n\nEnter your choice ::";

cin>>ch;

switch(ch)

{

case 1:

clrscr();

student::stud st;

cout<<"\nEnter the Student id ::" ;

cin>>id;

file.open("admission.dat",ios::in | ios::binary);

file.seekg(0,ios::beg);

while(file.read((char\*)&st,sizeof(student::stud)))

{

if(id==st.student\_id)

{

flag=1;

}

}

file.close();

if(flag==1)

{

file.open("fines.dat",ios::app | ios::binary);

fi.fine\_id=id;

cout<<"\nEnter type of fine(late,lost,lib,etc)::";

gets(fi.type);

cout<<"\nEnter total fine ::";

cin>>fi.fine;

file.write((char\*)&fi,sizeof(account::fine));

file.close();

}

else

{

cout<<"\n\n\n NO such id";

getch();

}

break;

case 2:

clrscr();

file.open("fines.dat", ios::in | ios::binary);

file1.open("temp.dat",ios::out | ios::binary);

file.seekg(0,ios::beg);

cout<<"Enter the Student id ::";

cin>>id;

while(file.read((char\*)&fi,sizeof(account::fine)))

{

if(id==fi.fine\_id)

{

flag=1;

cout<<"\n1.Student Id ::"<<fi.fine\_id;

cout<<"\n2.Type of Fine ::"<<fi.type;

cout<<"\n3.Total Fine ::"<<fi.fine;

do

{

cout<<"\n\nEnter the number written in front feild ::";

cin>>numb;

if(numb==1)

{

cout<<endl<<"Sorry It can't be changed ::";

}

else if(numb==2)

{

cout<<endl<<"New Fine Type(e.g late,lib)::";

gets(fi.type);

}

else if(numb==3)

{

cout<<endl<<"New Total fine ::";

cin>>fi.fine;

}

else

{

cout<<"\n\nWrong Number choice";

}

cout<<"\n\nWant to change any other(y|n)::";

ans=getche();

}while(ans=='y');

file1.write((char\*)&fi,sizeof(account::fine));

}

else

{

file1.write((char\*)&fi,sizeof(account::fine));

}

}

file.close();

file1.close();

if(flag==0)

{

cout<<"Record not found";

getch();

}

else

{

file.open("temp.dat",ios::in | ios::binary);

file1.open("fines.dat",ios::out | ios::binary);

file.seekg(0,ios::beg);

while(file.read((char\*)&fi,sizeof(account::fine)))

{

file1.write((char\*)&fi,sizeof(account::fine));

}

file.close();

file1.close();

remove("temp.dat");

}

break;

case 3:

clrscr();

cout<<"Enter the Student id ::" ;

cin>>id;

file.open("fines.dat",ios::in | ios::binary);

temp.open("temp.dat",ios::out | ios::binary);

file.seekg(0,ios::beg);

while(file.read((char\*)&fi,sizeof(account::fine)))

{

if(id==fi.fine\_id)

{

flag=1;

cout<<"\n\n Record deleted";

}

else

{

temp.write((char\*)&fi,sizeof(account::fine));

}

}

file.close();

temp.close();

if(flag==0)

{

cout<<"\n\n\nRecord not found";

}

else

{

file.open("fines.dat",ios::out | ios::binary);

temp.open("temp.dat",ios::in | ios::binary);

temp.seekg(0,ios::beg);

while(temp.read((char\*)&fi,sizeof(account::fine)))

{

file.write((char\*)&fi,sizeof(account::fine));

}

file.close();

temp.close();

remove("temp.dat");

}

getch();

break;

case 4:

clrscr();

cout<<"Enter the Student id ::" ;

cin>>id;

file.open("fines.dat",ios::in | ios::binary);

file.seekg(0,ios::beg);

while(file.read((char\*)&fi,sizeof(account::fine)))

{

if(id==fi.fine\_id)

{

flag=1;

cout<<"\nStudent Id ::"<<fi.fine\_id;

cout<<"\nType of Fine ::"<<fi.type;

cout<<"\nTotal Fine ::"<<fi.fine;

getch();

}

}

file.close();

if(flag==0)

{

cout<<"\n\n\nRecord not found";

getch();

}

break;

case 5:

this->account\_submenu();

break;

default:

cout<<"\n\n\nWrong Choice";

getch();

break;

}

}

}

//====================REVENUE================================================

void account::revenue\_account()

{

while(1)

{

clrscr();

char filename[25];

int cal=0;

fstream file,file1,temp;

int ch;

cout<<"1.Total Salary Paid"<<endl<<"2.Total Fee received"<<endl<<"3.Total Fine Received"

<<endl<<"4.Sub Main menu";

cout<<"\n\nEnter your choice ::";

cin>>ch;

switch(ch)

{

case 1:

clrscr();

account::salary sal;

cout<<"\nName of file should be year with month number eg(f2010\_1.dat) for your";

cout<<"\nconvinience";

cout<<"\n\n\nEnter the name of the file::";

gets(filename);

file.open(filename,ios::nocreate | ios::in | ios::binary);

if(!file)

{

cout<<"\n\n\Sorry Wrong name mentioned not available";

}

else

{

while(file.read((char\*)&sal,sizeof(account::salary)))

{

cal=cal+sal.basic+sal.hr+sal.mc-sal.pa;

}

cout<<"\n So the total Salary paid in this month is::";

cout<<cal;

file.close();

}

getch();

break;

case 2:

clrscr();

account::fee fe;

cout<<"\nName of file should be year with month number eg(s2010\_1.dat) for your";

cout<<"\nconvinience";

cout<<"\n\n\nEnter the name of the file::";

gets(filename);

file.open(filename,ios::nocreate | ios::in | ios::binary);

if(!file)

{

cout<<"\n\n\Sorry Wrong name mentioned not available";

}

else

{

while(file.read((char\*)&fe,sizeof(account::fee)))

{

cal=cal+fe.tuition+fe.bus+fe.magzine+fe.misc+fe.tour+fe.sport;

}

cout<<"\n\n\n So the total Fee Received in this month is::";

cout<<cal;

file.close();

}

getch();

break;

case 3:

clrscr();

account::fine fi;

file.open("fines.dat",ios::nocreate | ios::in | ios::binary);

while(file.read((char\*)&fi,sizeof(account::fine)))

{

cal=cal+fi.fine;

}

cout<<"\n So the total fine received in this month is::";

cout<<cal;

file.close();

getch();

break;

case 4:

this->account\_submenu();

break;

default:

cout<<"\n\n\nWrong Choice";

getch();

break;

}

}

}

//------------------------------------------------MENU CLASS FUNCTION--------

//====================MENU TIMES=============================================

void menu::times()

{

time=time+1;

}

//==================== MENU START============================================

void menu::start()

{

int ch=0;

while(1)

{

clrscr();

cout<<"1.Student Record"<<endl<<"2.Examination"<<endl<<"3.Account office"

<<endl<<"4.Miscellaneous [sport,staff,teacher,etc]"<<endl<<"5.School History and Information"<<endl<<"6.Exit";

cout<<"\nEnter your choice:: ";

cin>>ch;

switch(ch)

{

case 1:

student dent;

dent.student\_submenu();

break;

case 2:

exam exa;

exa.mark\_exam();

break;

case 3:

account acc;

acc.account\_submenu();

break;

case 4:

miscell mis;

mis.misc\_submenu();

break;

case 5:

ifstream file;

clrscr();

const int max=80;

char ch[80];

file.open("history.dat",ios::in | ios::binary);

while(file)

{

file.getline(ch,max);

cout<<ch<<endl;

}

getch();

break;

case 6:

exit(0);

break;

default:

cout<<"\nWrong choice";

getch();

}

}

}

//====================START PASSWORD=========================================

void menu::password()

{

menu menu\_obj1;

char ch,id[25],pass[25];

int i=0;

clrscr();

if(time==0)

{

menu\_obj1.times();

gotoxy(32,15);

cout<<"Made By::-HARNEET SINGH SETHI";

gotoxy(32,18);

cout<<"DOEACC 'A' LEVEL ID::- 556419";

gotoxy(32,23);

cout<<"USER ID (Project):: ";

gotoxy(32,25);

cout<<"PASSWORD (Doeacc):: ";

gotoxy(52,23);

while (1)

{

ch=getch();

if (ch == 8) //is backspace '\b'

{

if (i!=0)

{

id[--i]=0;

putch('\b');

putch(' ');

putch('\b');

}

}

if (isalpha(ch) || isdigit(ch))

{

id[i]=tolower(ch);

i++;

putch('\*');

}

if (ch==13)

{

id[i]=NULL; //null term

break;

}

}

i=0;

ch=NULL;

gotoxy(52,25);

//cout<<"PASSWORD (Doeacc):: ";

while (1)

{

ch=getch();

if (ch == 8) //is backspace '\b'

{

if (i!=0)

{

pass[--i]=0;

putch('\b');

putch(' ');

putch('\b');

}

}

if (isalpha(ch) || isdigit(ch))

{

pass[i]=tolower(ch);

i++;

putch('\*');

}

if (ch==13)

{

pass[i]=NULL; //null term

break;

}

}

if((strcmp(id,"project")==0)&&(strcmp(pass,"doeacc")==0))

{

menu\_obj1.start();

}

else

{

cout<<"\n\n\n\t\t\t Wrong password";

getch();

exit(1);

}

}

else

{

menu\_obj1.start();

}

}

void main()

{

char id[25],pass[25],ch;

menu m;

m.password();

}

//%%%%%%%%%%%%%%%%%%%%NORMAL SUBMENUS========================================

//%%%%%%%%%%%%%%%%%%%%NORMAL=================================================

//%%%%%%%%%%%%%%%%%%%%NORMAL=================================================

void student::student\_submenu()

{

int ch=0;

while(1)

{

clrscr();

cout<<"1.Admission"<<endl<<"2.TC"<<endl<<"3.Search/View"<<endl<<"4.Main Menu"

<<endl<<"5.Exit";

cout<<"\n\nEnter your choice ::";

cin>>ch;

switch(ch)

{

case 1:

this->admission();

break;

case 2:

this->transfer();

break;

case 3:

this->student\_search();

getch();

break;

case 4:

main();

break;

case 5:

exit(0);

default:

cout<<"\n\nWrong choice";

getch();

break;

}

}

}

void miscell::misc\_submenu()

{

int ch=0;

while(1)

{

clrscr();

cout<<"1.Teachers"<<endl<<"2.Sports"<<endl<<"3.Staff detail"<<endl<<"4.Main Menu"

<<endl<<"5.Exit";

cout<<"\n\nEnter your choice ::";

cin>>ch;

switch(ch)

{

case 1:

this->teacher\_member();

break;

case 2:

this->sport\_member();

break;

case 3:

this->staff\_member();

break;

case 4:

main();

break;

case 5:

exit(0);

break;

default:

cout<<"\n\nWrong choice";

getch();

break;

}

}

}

void account::account\_submenu()

{

int ch=0;

while(1)

{

clrscr();

cout<<"1.Salary"<<endl<<"2.Fee"<<endl<<"3.Fine"

<<endl<<"4.Total Revenue"<<endl<<"5.Main Menu"<<endl<<"6.Exit";

cout<<"\n\nEnter your choice ::";

cin>>ch;

switch(ch)

{

case 1:

this->salary\_account();

break;

case 2:

this->fee\_account();

break;

case 3:

this->fine\_account();

break;

case 4:

this->revenue\_account();

break;

case 5:

main();

break;

case 6:

exit(0);

break;

default:

cout<<"\n\nWrong choice";

getch();

break;

}

}

}

**SYSTEM IMPLEMEMTATION**

This phase is primarily concerned with user training, site preparation and file conversion. The candidate system is linked to terminals or remote sites, the telecommunication network and tests of the network and tests of the network along with the system are also included under implementation. System testing checks the read lines and accuracy of the system to access update and retrieve data from the files .Once the programs become available, test data are read and processed against the test files. If the successful, the program is then run with live data. Otherwise, a diagnostic procedure is used to locate and correct errors in the program.

**PROJECT TESTING**

A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements. A strategy must provide guidance for the practitioner and a set of milestones for the manager because the steps of the test strategy occur at a time when deadline pressure begins to rise, progress must be measurable and problems must surface as earl as possible.

**Following testing techniques are well known and the same strategy is adopted during this project testing.**

**Unit testing:**

Unit testing focuses verification effort on the smallest unit of software design- the software component or module. The unit test is white-box oriented. The module interface is tested to ensure that information properly flows into and of the program unit under test the local data structure has been examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm’s execution. Boundary conditions are tested to ensure that the module operated properly at boundaries established to limit or restrict processing. All independent paths through the control structure are exercised to ensure that all statements in a module haven executed at least once.

**Integration testing:**

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective of this test is to take unit tested components and build a program structure that has been dictated by design.

**Validation testing:**

At the culmination of integration testing, software is completely assembled as a package, interfacing errors have been uncovered and corrected, and a final series of software tests—validation testing-may begins. Validation can be defined in many ways, but a simple definition is that validation succeeds when software functions in a manner that can be reasonably expected by the customer.

**System testing:**

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Below we have described the two types of testing which have been taken for this project.

**Security testing:**

Any computer-based system that manages sensitive information causes actions that can improperly harm (or benefit) individuals is a target for improper or illegal penetration. Penetration spans a broad range of activities: hackers who attempt to penetrate system for sport; disgruntled employees who attempt to penetrate for revenge; dishonest individuals who attempt to penetrate for illicit personal gain.

For security purposes, when anyone who is not authorized user cannot penetrate this system. When programs first load it check for correct username and password. If any fails to act according will be simply ignored by the system.

**Performance Testing:**

Performance testing is designed to test the run-time performance of software Within the context of an integrated system. Performance testing occurs through out all steps in the testing process. Even at the unit level, the performance of an individual module may be assessed as white-box tests are conducted.

**Criteria for Completion of Testing**

Every time the customer/user executes a compute program, the program is being tested. This sobering fact underlines the importance of other software quality assurance activities.

**Validation Checks :**

Software testing is one element of broader topic that is often referred to as verification and validation. Verification refers to the set of activities that ensure that software correctly implements a specific function. Validation refers to a different set of activities that ensure that the software that has been built is traceable to customer requirements. Validation checks are useful when we specify the nature of data input. In this project while entering the data to many text box you will find the use of validation checks. When you try to input wrong data, your entry will be automatically abandoned.

**SYSTEM MAINTENANCE**

After the installation phase is completed and the user stall is adjusted to the challenges created by the candidate system, evaluation and the maintenance begin. Like any system, there is any aging process that requires periodic maintenance of hardware and software .If the new information is inconsistent with the design specifications, in tune with design specifications. The importance of maintenance is to continue to bring the new system to standards.

It has been estimated that maintenance of any software product usually requires much more efforts than the efforts necessary to develop the product.

Many study indicates that the relative effort of development of a typical system to its maintenance effort to roughly in the ratio 40:60.

**Maintenance involves performing any one or more of the following three kinds of activities:**

1. Correcting errors that were not discovered during the product-developing phase. This is called Corrective Maintenance.
2. Improving the Implementation of the system and enhancing the functionality of the system according to the customer’s requirements. This is called Perceptive Maintenance.
3. Porting the software to a new environment, e.g. to a new computer or to a new operating system. This is called Adaptive Maintenance.

**SYSTEM MAINTENANCE & DATA VALIDATION**

Before discussing the validation checks it is important to discuss the importance of the system security and maintenance as validation checks are a part of the system security features. There are basically two types of security associated with this system.

There are many problems associated with data security. Data duplicity being one of the major cancerous causing problems in identifying student/student. In various forms the student number which problem in identifying student. In various forms the student member which is unique for each student is used and displaced, but for changes on this field have been allowed in these forms so as to maintain the data integrity and avoid any duplicity. Some concept is being used for all other fields which are unique in nature as reference number is personal library, case number in case details etc.

Another problem which is found regarding the data security is that the loss of data due to improper assignment of fields size and type. To avoid this problem proper field size and type according to the maximum requirement is assigned.

**The other two basic problems associated with data security are:**

**1.** Identification : For user id

**2.** Authentication : System checking for password particular identification.

**3.** Authorization : System permits the resource to the authorized person. access control mechanism to prevent authorized losing to the system.

COMPUTER SECURITY

The School Management System is written is CPP language since the database designing and implementation is done using C++ file handling.

It is a secured system, as the system uses the binary mode of the file handling which is a good approach as any file created by this is not readable by the user it is not a plain text file. It can be read by the system, which has created this file.

As per this feature the data stored by the system is secured and most importantly the password and id check which is their in the software makes it free from any kind of mishap as only authorized person can only enter into the system.

FUTURE PROSPECTS

This School Management System is to replace the existing manual School Management System where large amount of paper work is required even for a small school. In my opinion practically and economically this system is always needed in the small of large financial educational organizations or in office to assist the employees.

The system is able to calculate the fine, fees of student but more validation and accuracy can be put in so that the employees can completely rely on it .It can also be used to find out total revenue i.e. amount has been received as the fine or fees and also total amount paid in form of salary.

The Project or this software can be used in any organization for the better communication. More and more corporation’s information assets can be entered and use in different levels. Adding new features for the better communication in the intranet is the purpose of the system. In this regards, it can be integrated with any office communication software for better and effective interoffice, departmental knowledge and problem sharing. This will as a single platform to solve multitude of problem of an institute/organization. Making system more effective and intelligent we can add more feature in the future like chat, voice, shared notice board, and tools for the quick identification of the users problem.

USER MANUAL

First the user have to enter the password and user id at the time of starting of the software where user id is project and password is doeacc, it can be written both in upper and lower case that won’t make any difference.

On the commencement of the software the main menu will appear which will have Student record, Examination, Account office, Miscellaneous and School history. Under the Student Record menu item comes admission, Transfer certificate, Search Student. Now under the admission sub item comes student information such as registration of student, removal and update of student record, while under the Transfer Certificate comes the providing, removal, update, view of transfer certificate and finally under the search we can view student record.

Second menu item in main menu is the Examination under this item comes sub menu i.e. enter, view, update, class wise view and removal of student marks.

Third menu item in main menu is the Account office under this menu item contains another sub menu which contains item such as Salary, Fee, Fine and Total revenue items where Salary consist of all information regarding salary paid to the staff members, under the salary sub menu comes enter, update, removal and view of the salary same with the Fee contain all information about fee received by the student, it also have submenu consist enter, update, removal and view of the student fee information and under Fine again comes sub menu having items like view and storing of all fine information and under the Total revenue item have submenu for viewing of total salary paid or total fee, fine received.

Fourth menu item in main menu is the Miscellaneous under which menu item we have another sub menu which contains item such as Teacher, Staff and Sport items where Teacher item consist of all information regarding teacher information, under the Teacher we have a sub menu which allow us enter, update, removal and view of the teacher information same with the staff contain all information about staff, there is again a submenu having items like enter, update, removal and view of the staff and under the sport item sub menu we have same item like enter, update, removal and view of sport information for storing and managing the information.

The last menu item is the School history containing a text page having past history and information about the school foundation.

CONCLUSION

After prepared analysis and design for this system, we try to replace the manual system by automated one. In our analysis we used high techniques to prepare a good design to be easy for implementation phase, so we hope that this system will appear in a good case. This new system is required to match new fast technology age for helping school management requirement. The new system can be integrated with other school management systems to get full benefit.

BIBLIOGRAPHY

**For the purpose of designing the above project the following books have been referred:**

**Book Name** **Author Name**

1. Object Oriented programming & C++ E.Balagurusamy
2. Programming in ANSI C 4E E.Balagurusamy
3. Let Us C++ Yashwant Kanitkar
4. Documentation Rajib Mall