

# **SUBJECT WISE STUDENT ANALYSIS**

*A*

*Mini Project Report*

*Submitted in partial fulfilment of the  
Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

**IN**

**INFORMATION TECHNOLOGY**

**By**

**NIKITHA 1602-17-737-087**

**CHAITANYA 1602-17-737-095**

**SANDEEP 1602-17-737-101**



**Department of Information Technology  
Vasavi College of Engineering (Autonomous)  
(Affiliated to Osmania University)**

Ibrahimbagh, Hyderabad-31

2018

**Vasavi College of Engineering (Autonomous)  
(Affiliated to Osmania University)  
Hyderabad-500 031  
Department of Information Technology**



## **DECLARATION BY THE CANDIDATES**

We, NIKITHA, CHAITANYA, SANDEEP, bearing hall ticket numbers, 1602-17-737-087, 1602-17-737-095, 1602-17-737-101 respectively, hereby declare that the project report entitled "SUBJECT WISE STUDENT ANALYSIS" Department of Information Technology, Vasavi College of Engineering, Hyderabad, is submitted in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering in Information Technology**

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

**NIKITHA  
1602-17-737-087  
CHAITANYA  
1602-17-737-095  
SANDEEP  
1602-17-737-101**

(Faculty In-Charge)

(Head, Dept of IT)

# CONTENTS

1.Introduction.....	1
1.1 Purpose.....	2
1.2 Scope.....	2
2.Project description.....	2
3.Technology.....	3
3.1 Overview of technology used.....	3
3.2 Software Requirements.....	4
4.Code Templates.....	4
MODULE 1.....	15
MODULE 2.....	16
MODULE 3.....	17
MODULE 4.....	18
MODULE 5.....	19
MODULE 6.....	19
MODULE 7.....	24
MODULE 8.....	24
5.Output Screens.....	25
6.Conclusion and Future Scope.....	27

# 1.Introduction

Our project gives the complete description of the student analysis in academics. The project is different from normal other projects of attendance checking. This project tells how student attendance deals with his academic percentage.

This project completely deals how student performs and his regularity, which describes his attitude in time making. This project also has a function of classification of all students based on their academics.

This project does exist in today's market but it is not so reachable to every teacher. A school or a college with basic infrastructure and few computers can use this project in an easy manner to store data. It is very simple to use and get command on the outputs in our project.

In this project it tells us how attendance does matter of getting good score in academics. In some cases there might be few students whose score in academics doesn't depend upon attendance but it still matters to a lot of them. It is helpful for students to analyze themselves and come with good score with a good percentage of attendance.

Here the complete data of the student is very secured and protected .No one can access it simply .To access the data in our project it needs a secret code which to only the respective teacher related to that branch.

## **1.1 Purpose**

We see that it been though for teacher to sort all students within a class of 100+ students in cooperative education system.so it helps teacher each and every students based on academic their score , attendance and different classification which we use in our project.

This project will not only be helping the teacher but students also to their marks and evaluate themselves based on their on their score in academics and attendance, which matters a lot student.

## **1.2 Scope**

Our project do exists in today's world but we want to make an advanced one of this by adding more information which is apart of academic score and attendance like the complete academic syllabus with divisions of day to day.Giving a complete descriptive perspective on the student based on their attiudue which teacher observe and scorces in each subject will persepective attendance in that subject.

## **2.Project description**

Our project "Demonstrates" a 'SUBJECT WISE STUDENT ANALYSIS' using C++

language. The project takes different inputs form for display information stored in it.It uses searching technique for bringing out the information which stored in files.Here different files are used named as CSE,EEE,MEC,CIV,ECE,IT.In different files the details of the students related that branch are stored here.The details stored here are students name,roll\_no,percentage,attendance and etc.Few of the inputs are given the user to display the complete information of required one.

### 3.Technology

#### 3.1 Overview of technology used

C++ is an object-oriented language based on the C programming language. It can be viewed as a superset of C. Almost all of the features and constructs available in C are also available in C++. However, C++ is more than just an extension of C. Its additional features support the programming style known as *object-oriented programming*. Several features that are already available in C, such as input and output may be implemented differently in C++. In C++ you may use the conventional C input and output routines or you may use object oriented input and output by using the I/O Stream class library.

C++ was developed by Bjarne Stroustrup of AT&T Bell Laboratories. It was originally based on the definition of the C language stated in *The C Programming Language* by Brian W. Kernighan and Dennis M. Ritchie. This C language definition is commonly called *K&R C*. Since then, the International Standards Organization C language definition (referred to here as ISO/ANSI C ) has been approved. It specifies many of the features that K&R left unspecified. Some features of ISO/ANSI C have been incorporated into the current definition of C++, and some parts of the ISO/ANSI C definition have been motivated by C++.

While there is currently no C++ standard comparable to the ISO/ANSI C definition, an ISO committee is working on such a definition. The draft of

the Working Paper for Draft Proposed American National Standard for Information Systems - Programming Language C++ , X3J16/92-0091, is the base document for the ongoing standardization of C++. The IBM C and C++ Compilers adheres to the version of the ISO/ANSI working paper dated September 17, 1992.

### 3.2 Software Requirements

Any basic C++ compiler such a G++ compiler or Code Blocks software .

## 4.Code Templates

- The "SUBJECT WISE STUDENT ANALYSIS", a program which encloses all modules in it as

```
#include<iostream>
#include<iomanip>
#include<fstream>
#include<string.h>
#include<conio.h>
using namespace std;
fstream fp;
class subject
{
    public:
    float marks;
    float att;
};
class student
{
    int n;
    public:
```

```

        string name;
        subject m[6];
        int roll;
        string password;
        void show_details()
        {
            cout<<"\n*****
*****";
            cout<<"\n ROLL NUMBER:"<<roll;
            cout<<"\n NAME:"<<name;
            cout<<"\nDEPARTEMENT CODE"<<n;
                cout<<"\n MARKS AND ATTENDANCE ARE"<<endl;
                for(int i=0;i<6;i++)
                {
                    cout<<setw(5)<<m[i].marks<<setw(5)<<m[i].att<<endl;
                }
            }
        void read()
        {
            int i;float a;float b;
            cout<<"enter the roll number of the student:";
            cin>>roll;
            cout<<"enter the name of the student:";
            cin>>name;
            cout<<"enter the prescribed password of the student:";
            cin>>password;
            cout<<"enter the departement code:";
            cin>>n;
                for(i=0;i<6;i++)
                {
                    printf("enter the marks and attendance of %d sub",i+1);
                    scanf("%f%f",&a,&b);
                    m[i].marks=a;
                    m[i].att=b;
                }
            }
        };
        student stud;
        void write_s(int n)
        {
            int ch;
            do
            {

```



```

switch(n)
{
    case 1:{fp.open("cse.dat",ios::app);}
    break;
    case 2:{fp.open("civil.dat",ios::app);}
    break;
    case 3:{fp.open("ece.dat",ios::app);}
    break;
    case 4:{fp.open("eee.dat",ios::app);}
    break;
    case 5:{fp.open("mec.dat",ios::app);}
    break;
    case 6:{fp.open("it.dat",ios::app);}
    break;
}
    stud.read();
    fp.write((char *)&stud,sizeof(student));
    cout<<"do you wanna enter more details:";
    cout<<"1.YES 2. NO";
    cin>>ch;
    fp.close();
}while(ch!=2);
    fp.close();
}
void teacher_display(int n)
{
    cout<<"THE DETAILS OF STUDENTS:";
    switch(n)
    {
        case 1:{fp.open("cse.dat",ios::in);}
        break;
        case 2:{fp.open("civil.dat",ios::in);}
        break;
        case 3:{fp.open("ece.dat",ios::in);}
        break;
        case 4:{fp.open("eee.dat",ios::in);}
        break;
        case 5:{fp.open("mec.dat",ios::in);}
        break;
        case 6:{fp.open("it.dat",ios::in);}
        break;
    }
    while(fp.read((char *)&stud,sizeof(student)))
    {
        stud.show_details();
    }
}

```

```

        cout<<endl<<endl;
    }
    fp.close();
}
void display_sp(int num,int d)
{
    int flag=0;
    switch(d)
    {
        case 1:{fp.open("cse.dat",ios::in);}
            break;
        case 2:{fp.open("civil.dat",ios::in);}
            break;
        case 3:{fp.open("ece.dat",ios::in);}
            break;
        case 4:{fp.open("eee.dat",ios::in);}
            break;
        case 5:{fp.open("mech.dat",ios::in);}
            break;
        case 6:{fp.open("it.dat",ios::in);}
            break;
    }
    if(!fp)
        cout<<"unable to open the file";
    while(fp.read((char *)&stud,sizeof(student)))
    {

        if(num==stud.roll)
        {

            stud.show_details();
            flag=1;

        }
    }
    if(flag==0)
        cout<<"\n record not found";
    fp.close();
}
void modify(int num,int d)
{
    switch(d)
    {
        case 1:{fp.open("cse.dat",ios::out|ios::in);}
            break;
        case 2:{fp.open("civil.dat",ios::out|ios::in);}

```

```

        break;
    case 3:{fp.open("ece.dat",ios::out|ios::in);}
        break;
    case 4:{fp.open("eee.dat",ios::out|ios::in);}
        break;
    case 5:{fp.open("mec.dat",ios::out|ios::in);}
        break;
    case 6:{fp.open("it.dat",ios::out|ios::in);}
        break;
    }
    while(fp.read((char *)&stud,sizeof(student)))
    {
        if(stud.roll==num)
        {
            stud.show_details();
            cout<<"enter the new details ";
            stud.read();
            int pos=(-1)*sizeof(student);
            fp.seekp(pos,ios::cur);
            fp.write((char *)&stud,sizeof(student));
        }
    }
    fp.close();
}
int a[10],b[10];
void lesst(int p,int s,int d)
{
    int count=0;int i=0;
    switch(d)
    {
        case 1:fp.open("cse.dat",ios::in);
        break;
        case 2:fp.open("civil.dat",ios::in);
        break;
        case 3:fp.open("eee.dat",ios::in);
        break;
        case 4:fp.open("ece.dat",ios::in);
        break;
        case 5:fp.open("mec.dat",ios::in);
        break;
        case 6:fp.open("it.dat",ios::in);
        break;
    }
    while(fp.read((char *)&stud,sizeof(student)))

```

```

    {
        if(stud.m[s].marks<p)
        {
            b[i++]=stud.m[s].marks;
            a[i++]=stud.m[s].att;
            cout<<endl<<stud.roll<<setw(6)<<stud.m[s].att;
            count++;
        }
    }
    cout<<"\nThe number of students who scaored less
than"<<p<<"are"<<count<<"\n";
    for(int j=0;j<i;j++)
    {
        cout<<b[j];
        for(int k=a[j]/10;k>=0;k--)
        {
            cout<<" *";
        }
        cout<<"\n\n";
    }
    fp.close();
}

void greatt(int p,int s,int d)
{
    int i;
    int count=0;
    switch(d)
    {
        case 1:fp.open("cse.dat",ios::in);
        break;
        case 2:fp.open("civil.dat",ios::in);
        break;
        case 3:fp.open("eee.dat",ios::in);
        break;
        case 4:fp.open("ece.dat",ios::in);
        break;
        case 5:fp.open("mec.dat",ios::in);
        break;
        case 6:fp.open("it.dat",ios::in);
        break;
    }
}

```

```

    }
    while(fp.read((char *)&stud,sizeof(student)))
    {
        if(stud.m[s].marks>p)
        {
            b[i++]=stud.m[s].marks;
            a[i++]=stud.m[s].att;
            cout<<endl<<stud.roll<<setw(6)<<stud.m[s].att;
            count++;
        }
    }
    cout<<"\n\nThe number of students who scored marks greater than
"<<p<<"are"<<count<<"\n";
    for(int j=0;j<i;j++)
    {
        cout<<b[j];
        for(int k=a[j]/10;k>=0;k--)
        {
            cout<<" *";
        }
        cout<<"\n\n";
    }

    fp.close();
}

void equalt(int p,int s,int d)
{
    int i;
    int count=0;
    switch(d)
    {
        case 1:fp.open("cse.dat",ios::in);
        break;
        case 2:fp.open("civil.dat",ios::in);
        break;
        case 3:fp.open("eee.dat",ios::in);
        break;
        case 4:fp.open("ece.dat",ios::in);
        break;
        case 5:fp.open("mec.dat",ios::in);
        break;
        case 6:fp.open("it.dat",ios::in);
        break;
    }
}

```

```

    }
    while(fp.read((char *)&stud,sizeof(student)))
    {
        if(stud.m[s].marks==p)
        {
            b[i++]=stud.m[s].marks;
            a[i++]=stud.m[s].att;
            cout<<endl<<stud.roll<<setw(6)<<stud.m[s].att;
            count++;
        }
    }

    cout<<"\nThe number of students who scored "<<p<<"are"<<count<<"\n";
    for(int j=0;j<i;j++)
    {

        cout<<b[j];
        for(int k=a[j]/10;k>=0;k--)
        { cout<<" *";
        }
        cout<<"\n\n";

    }
    fp.close();
}

void fault(int p,int s,int d)
{
    int i;
    int count=0;
    switch(d)
    {
        case 1:fp.open("cse.dat",ios::in);
        break;
        case 2:fp.open("civil.dat",ios::in);
        break;
        case 3:fp.open("eee.dat",ios::in);
        break;
        case 4:fp.open("ece.dat",ios::in);
        break;
        case 5:fp.open("mec.dat",ios::in);
        break;
        case 6:fp.open("it.dat",ios::in);
        break;
    }
}

```

```

while(fp.read((char *)&stud,sizeof(student)))
{
    if(stud.m[s].marks<14)
    {
        b[i++]=stud.m[s].marks;
        a[i++]=stud.m[s].att;

        cout<<endl<<stud.roll<<setw(6)<<stud.m[s].att;
        count++;
    }
}
cout<<"\nThe total number of failures are"<<count<<"\n";
for(int j=0;j<i;j++)
{

    cout<<b[j];

    for(int k=a[j]/10;k>=0;k--)
    {
        cout<<" *";
    }
    cout<<"\n\n";
}
fp.close();
}
void sub()
{
    int l,s,p,d;
    cout<<"\nenter your department \n 1.CSE \n 2.CIVIL \n 3.EEE \n 4.ECE \n
5.MECH \n6.IT";
    cin>>d;
    cout<<"\nenter your desired subject";
    cin>>s;
    cout<<"\n enter the marks";
    cin>>p;
    cout<<"\n enter your choice\n 1.less than \n 2.greater than\n3. equal \n4. fail";
    cin>>l;
    switch(l)
    {
        case 1:lesst(p,s,d);
        break;
        case 2: greatt(p,s,d);
        break;
        case 3: equalt(p,s,d);
        break;
    }
}

```

```

        case 4: fault(p,s,d);
        break;
    }
}
void teach_menu(int d)
{
    int ch;
    int num;
    do
    {
        cout<<"\nenter your choice";
        cout<<"\n1.create a new record of the student";
        cout<<"\n2.display all the records";
        cout<<"\n3.search student record";
        cout<<"\n4.edit student record";
        cout<<"\n5.subject wise analysis:";
        cout<<"\n6.exit";
        cin>>ch;
        switch(ch)
        {

            case 1:write_s(d);
                break;
            case 2:teacher_display(d);
                break;
            case 3:cout<<"enter the roll number you wanna search";
                cin>>num;
                display_sp(num,d);
                break;
            case 4:cout<<"enter the roll number";
                cin>>num;
                modify(num,d);
                break;
            case 5:sub();
                break;
            case 6:break;
        }
    }while(ch!=6);
}
int check_pass(int dep,int rol,string s)
{
    switch(dep)
    {
        case 1:{fp.open("cse.dat",ios::in);}
        break;
    }
}

```



```

        case 2:{fp.open("civil.dat",ios::in);}
            break;
        case 3:{fp.open("ece.dat",ios::in);}
            break;
        case 4:{fp.open("eee.dat",ios::in);}
            break;
        case 5:{fp.open("mec.dat",ios::in);}
            break;
        case 6:{fp.open("it.dat",ios::in);}
            break;
    }
    while(fp.read((char *)&stud,sizeof(student)))
    {
        if(stud.roll==rol)
        {
            if(stud.password==s)
            {
                stud.show_details();
                return 1;
            }
        }
    }
    return 0;
}
int main()
{
    int i;int ch;
    string sec;
    sec="cultic";
    int a,b,c;int m;
    string r;

    cout<<"_____".
    _____,
    cout<<"\nEnter your designation";
    cout<<"\n1.TEACHER \n 2.STUDENT";
    cin>>ch;
    switch(ch)
    {
        case 1:{
            cout<<"enter your department";

            cout<<"\n1.CSE\n2.CIVIL\n3.ECE\n4.EEE\n5.MECHANICAL\n6.IT";
            cin>>a;
            for(i=0;i<3;)

```

```

        {
            cout<<"\nenter your secret code";
            cin>>r;
            if(r!=sec)
            {
                printf("\nyou have entered it wrong please try
again");
                i++;
                cout<<"\n You have "<<3-i<<"chances
more.....";
            }
            else break;
        }
        if(i==3)
            cout<<"\nyou have entered wrong
password.....\naborting";
            if(i<3)
                teach_menu(a);
            }
            break;
        case 2:
            {cout<<"\n HELLO STUDENT!!!!!!welcome to subject wise
student analysis";
                cout<<"\n Enter your department:";

                cout<<"\n1.CSE\n2.CIVIL\n3.ECE\n4.EEE\n5.MECHANICAL\n6.IT";
                cin>>a;
                cout<<"\nenter your roll number:";
                cin>>b;
                cout<<"\nenter your password";
                cin>>r;
                if(!check_pass(a,b,r))
                    cout<<"\n PASSWORD INCORRECT\n please enter
appropriate password";
            }
        }

        return 0;
    }

```

## MODULE 1

- In this section we will be reading complete input details of student.

```
void read()
{
    int i;float a;float b;
    cout<<"enter the roll number of the student:";
    cin>>roll;
    cout<<"enter the name of the student:";
    cin>>name;
    cout<<"enter the prescribed password of the student:";
    cin>>password;
    cout<<"enter the departement code:";
    cin>>n;
    for(i=0;i<6;i++)
    {
        printf("enter the marks and attendance of %d
sub",i+1);
        scanf("%f%f",&a,&b);
        m[i].marks=a;
        m[i].att=b;
    }
};
```

## MODULE 2

- This module encloses code of files which we will be using to access data in files.

```
void write_s(int n)
{
    int ch;
    do
    {
        switch(n)
        {
            case 1:{fp.open("cse.dat",ios::app);}
        }
    }
};
```

```

        break;
        case 2:{fp.open("civil.dat",ios::app);}
        break;
        case 3:{fp.open("ece.dat",ios::app);}
        break;
        case 4:{fp.open("eee.dat",ios::app);}
        break;
        case 5:{fp.open("mec.dat",ios::app);}
        break;
        case 6:{fp.open("it.dat",ios::app);}
        break;
    }
    stud.read();
    fp.write((char *)&stud,sizeof(student));
    cout<<"do you wanna enter more details:";
    cout<<"1.YES 2. NO";
    cin>>ch;
    fp.close();
}while(ch!=2);
    fp.close();
}

```

### MODULE 3

- This module encloses the code of teachers display. The output required for the teacher is shown here. From files information is extracted.

```

void teacher_display(int n)
{
    cout<<"THE DETAILS OF STUDENTS:";
    switch(n)
    {
        case 1:{fp.open("cse.dat",ios::in);}
        break;
        case 2:{fp.open("civil.dat",ios::in);}
        break;
        case 3:{fp.open("ece.dat",ios::in);}
        break;
        case 4:{fp.open("eee.dat",ios::in);}
        break;
        case 5:{fp.open("mec.dat",ios::in);}
    }
}

```

```

        break;
        case 6:{fp.open("it.dat",ios::in);}
        break;
    }
    while(fp.read((char *)&stud,sizeof(student)))
    {
        stud.show_details();
        cout<<endl<<endl;
    }
    fp.close();
}

```

## MODULE 4

- This module encloses the code of students display. Here the information required for student is only shown. From files information is extracted.

```

void display_sp(int num,int d)
{
    int flag=0;
    switch(d)
    {
        case 1:{fp.open("cse.dat",ios::in);}
        break;
        case 2:{fp.open("civil.dat",ios::in);}
        break;
        case 3:{fp.open("ece.dat",ios::in);}
        break;
        case 4:{fp.open("eee.dat",ios::in);}
        break;
        case 5:{fp.open("mech.dat",ios::in);}
        break;
        case 6:{fp.open("it.dat",ios::in);}
        break;
    }
    if(!fp)
        cout<<"unable to open the file";
    while(fp.read((char *)&stud,sizeof(student)))
    {
        if(num==stud.roll)
        {

```

```

                                stud.show_details();
                                flag=1;
                            }
                        }
                    if(flag==0)
                        cout<<"\n record not found";
                    fp.close();
                }

```

## MODULE 5

- This module encloses the code of modify function where information in files are modified. This function is only available for teachers.

```

void modify(int num,int d)
{
    switch(d)
    {
        case 1:{fp.open("cse.dat",ios::out|ios::in);}
                break;
        case 2:{fp.open("civil.dat",ios::out|ios::in);}
                break;
        case 3:{fp.open("ece.dat",ios::out|ios::in);}
                break;
        case 4:{fp.open("eee.dat",ios::out|ios::in);}
                break;
        case 5:{fp.open("mec.dat",ios::out|ios::in);}
                break;
        case 6:{fp.open("it.dat",ios::out|ios::in);}
                break;
    }
    while(fp.read((char *)&stud,sizeof(student)))
    {
        if(stud.roll==num)
        {
            stud.show_details();
            cout<<"enter the new details ";
            stud.read();
            int pos=(-1)*sizeof(student);
            fp.seekp(pos,ios::cur);

```

```

        fp.write((char *)&stud,sizeof(student));
    }
}
fp.close();
}

```

## MODULE 6

- This module encloses data which is to be searched. This functions can be used only by teachers. The searching technique has been classified into 4 types.

lesst

greatt

equilat

failt

```

void lesst(int p,int s,int d)
{
    int count=0;int i=0;
    switch(d)
    {
        case 1:fp.open("cse.dat",ios::in);
        break;
        case 2:fp.open("civil.dat",ios::in);
        break;
        case 3:fp.open("eee.dat",ios::in);
        break;
        case 4:fp.open("ece.dat",ios::in);
        break;
        case 5:fp.open("mec.dat",ios::in);
        break;
        case 6:fp.open("it.dat",ios::in);
        break;
    }
    while(fp.read((char *)&stud,sizeof(student)))
    {
        if(stud.m[s].marks<p)
        {
            b[i++]=stud.m[s].marks;

```

```

                a[i++]=stud.m[s].att;
                cout<<endl<<stud.roll<<setw(6)<<stud.m[s].att;
                count++;
            }
        }
        cout<<"\n\nThe number of students who scaored less
than"<<p<<"are"<<count<<"\n";
        for(int j=0;j<i;j++)
        {

                cout<<b[j];
                for(int k=a[j]/10;k>=0;k--)
                {
                        cout<<" *";
                }
                cout<<"\n\n";

        }
        fp.close();
}
void greatt(int p,int s,int d)
{
        int i;
        int count=0;
        switch(d)
        {
                case 1:fp.open("cse.dat",ios::in);
                break;
                case 2:fp.open("civil.dat",ios::in);
                break;
                case 3:fp.open("eee.dat",ios::in);
                break;
                case 4:fp.open("ece.dat",ios::in);
                break;
                case 5:fp.open("mec.dat",ios::in);
                break;
                case 6:fp.open("it.dat",ios::in);
                break;
        }
        while(fp.read((char *)&stud,sizeof(student)))
        {
                if(stud.m[s].marks>p)

```



```

        {
            b[i++]=stud.m[s].marks;
            a[i++]=stud.m[s].att;
            cout<<endl<<stud.roll<<setw(6)<<stud.m[s].att;
            count++;
        }
    }
    cout<<"\n\nThe number of students who scored marks greater than
"<<p<<"are"<<count<<"\n";
    for(int j=0;j<i;j++)
    {
        cout<<b[j];
        for(int k=a[j]/10;k>=0;k--)
        {
            cout<<" *";
        }
        cout<<"\n\n";
    }

    fp.close();
}
void equalt(int p,int s,int d)
{
    int i;
    int count=0;
    switch(d)
    {
        case 1:fp.open("cse.dat",ios::in);
        break;
        case 2:fp.open("civil.dat",ios::in);
        break;
        case 3:fp.open("eee.dat",ios::in);
        break;
        case 4:fp.open("ece.dat",ios::in);
        break;
        case 5:fp.open("mec.dat",ios::in);
        break;
        case 6:fp.open("it.dat",ios::in);
        break;
    }
    while(fp.read((char *)&stud,sizeof(student)))
    {
        if(stud.m[s].marks==p)

```

```

        {
            b[i++]=stud.m[s].marks;
            a[i++]=stud.m[s].att;
            cout<<endl<<stud.roll<<setw(6)<<stud.m[s].att;
            count++;
        }
    }

    cout<<"\nThe number of students who scored "<<p<<"are"<<count<<"\n";
    for(int j=0;j<i;j++)
    {
        cout<<b[j];
        for(int k=a[j]/10;k>=0;k--)
        { cout<<" *";
        }
        cout<<"\n\n";
    }
    fp.close();
}

void fault(int p,int s,int d)
{
    int i;
    int count=0;
    switch(d)
    {
        case 1:fp.open("cse.dat",ios::in);
        break;
        case 2:fp.open("civil.dat",ios::in);
        break;
        case 3:fp.open("eee.dat",ios::in);
        break;
        case 4:fp.open("ece.dat",ios::in);
        break;
        case 5:fp.open("mec.dat",ios::in);
        break;
        case 6:fp.open("it.dat",ios::in);
        break;
    }
    while(fp.read((char *)&stud,sizeof(student)))
    {
        if(stud.m[s].marks<14)
        {

```

```

        b[i++]=stud.m[s].marks;
        a[i++]=stud.m[s].att;

        cout<<endl<<stud.roll<<setw(6)<<stud.m[s].att;
        count++;
    }
}
cout<<"\n\nThe total number of failures are"<<count<<"\n";
for(int j=0;j<i;j++)
{

    cout<<b[j];

    for(int k=a[j]/10;k>=0;k--)
    {
        cout<<" *";
    }
    cout<<"\n\n";
}
fp.close();
}

```

## MODULE 7

- Here we will be taking inputs like deapatment,desired subject and marks.
- Based on given inputs we will using different sorting criteria and dispalyed.

```

void sub()
{
    int l,s,p,d;
    cout<<"\nenter your department \n 1.CSE \n 2.CIVIL \n 3.EEE \n 4.ECE \n
5.MECH \n6.IT";
    cin>>d;
    cout<<"\nenter your desired subject";
    cin>>s;
    cout<<"\n enter the marks";
    cin>>p;
    cout<<"\n enter your choice\n 1.less than \n 2.greater than\n3. equal \n4. fail";
    cin>>l;
    switch(l)
    {
        case 1:lesst(p,s,d);
        break;
        case 2: greatt(p,s,d);
        break;
    }
}

```

```

        case 3: equalt(p,s,d);
        break;
        case 4: failt(p,s,d);
        break;
    }
}

```

## MODULE 8

- Here the complete teachers menu are displayed. Teacher can choose any of them and access the data in it.

```

void teach_menu(int d)
{
    int ch;
    int num;
    do
    {
        cout<<"\nenter your choice";
        cout<<"\n1.create a new record of the student";
        cout<<"\n2.display all the records";
        cout<<"\n3.search student record";
        cout<<"\n4.edit student record";
        cout<<"\n5.subject wise analysis:";
        cout<<"\n6.exit";
        cin>>ch;
        switch(ch)
        {
            case 1:write_s(d);
                break;
            case 2:teacher_display(d);
                break;
            case 3:cout<<"enter the roll number you wanna search";
                cin>>num;
                display_sp(num,d);
                break;
            case 4:cout<<"enter the roll number";
                cin>>num;
                modify(num,d);
                break;
            case 5:sub();

```

```

        break;
    case 6:break;
    }
}while(ch!=6);
}

```

## 5.Output Screens

```

C:\WINDOWS\system32\cmd.exe - s

enter your choice
1.less than
2.greater than
3. equal
4. fail2

1 100
2 85
3 100
The number of students who scored marks greater than 20are3

30 *
0 * * * * * * * * * *
30 *
0 * * * * * * * * *
30 *
0 * * * * * * * * *

enter your choice
1.create a new record of the student
2.display all the records
3.search student record
4.edit student record
5.subject wise analysis:
6.exit_

```

```
Select C:\WINDOWS\system32\cmd.exe
*****
ROLL NUMBER:3
NAME:yeshwi
DEPARTEMENT CODE1
MARKS AND ATTENDANCE ARE
30 100
29 100
26 95
30 100
85 95
30 100

enter your choice
1.create a new record of the student
2.display all the records
3.search student record
4.edit student record
5.subject wise analysis:
6.exit5

enter your department
1.CSE
2.CIVIL
```

## 6.Conclusion and Future Scope

- It makes easy for teacher to sort out the students based on their regularity .The teacher can compare student attendance with their marks and can come to an conclusion about the student.The do exist now and we do hope of making advanced one of our project.