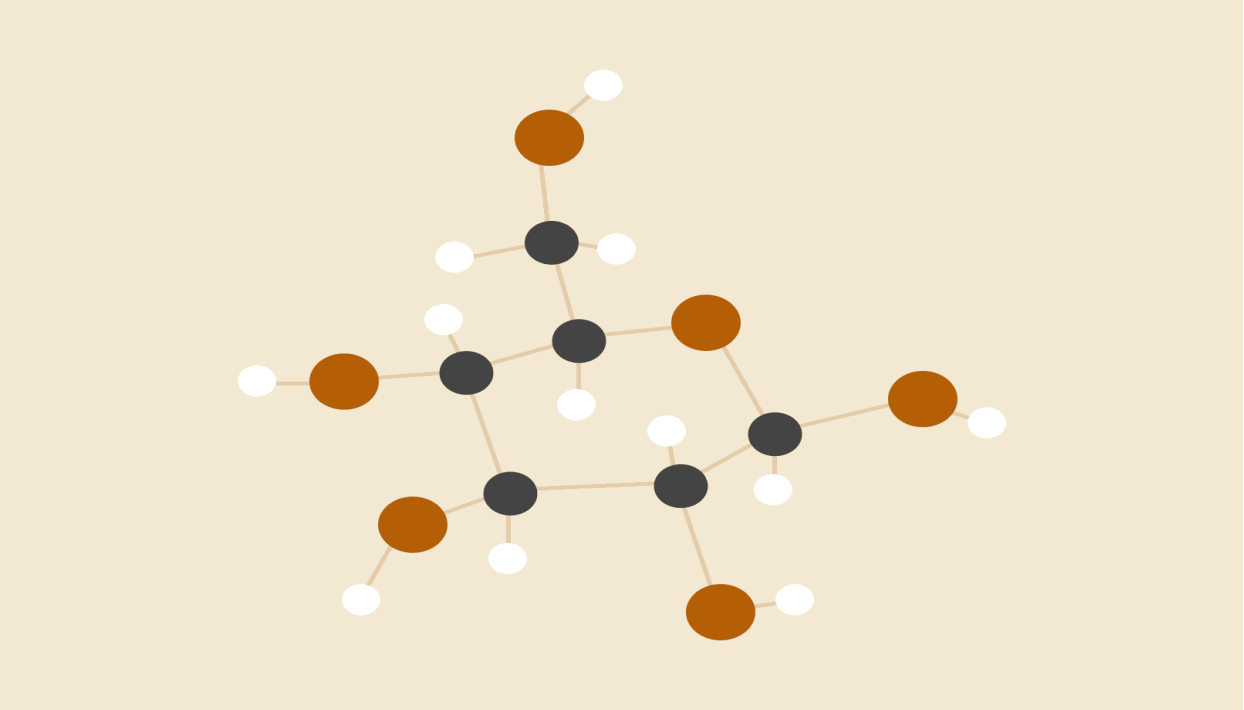
PROJECT REPORT

“Hospital Management System”



Group Details

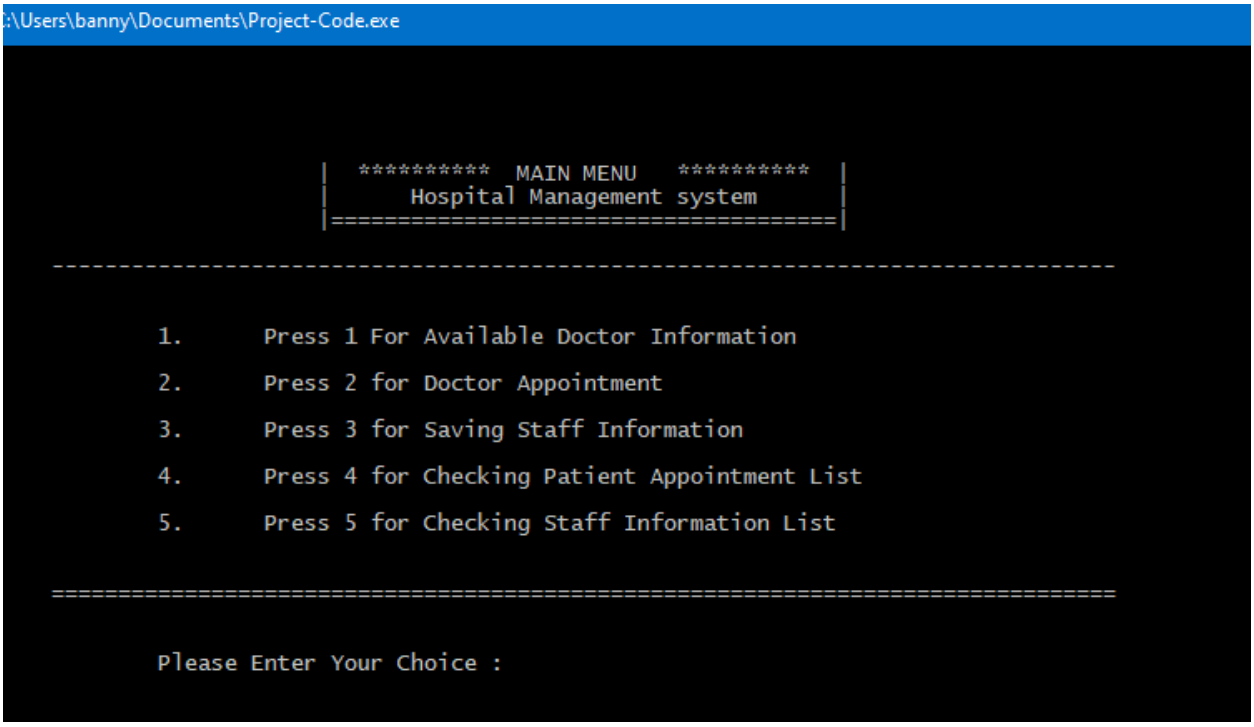
Name	Student ID
Sazia Ahmed	191014076
Anisur Rahman Tajmir	193014050
Farjahan Akter Boby	201014007
Apon Ghosh	201014069

Submission Date: 16-01-2021

Project Overview:

Hospital Management System is one of the most popular projects among students. So, we tried our level best to make one by ourselves.

In this project, we cover 1. Available Doctor list, 2. Make a Doctor's Appointment 3. Saving staff information 4. Checking patient appointment list 5. Checking Staff Information list. As an example, we already saved some doctors information like his/her specialty and consulting hours. By entering "1", we can see that information. By entering "2", we can make an appointment with any of those doctors. We can add new staff by filling some of their information in our hospital through entering number "3" whereas we can see patients' appointment list if we enter number "4". And the last but not the least one, "5", by which we can see our stuff information list.



Object-Oriented Programming Concepts

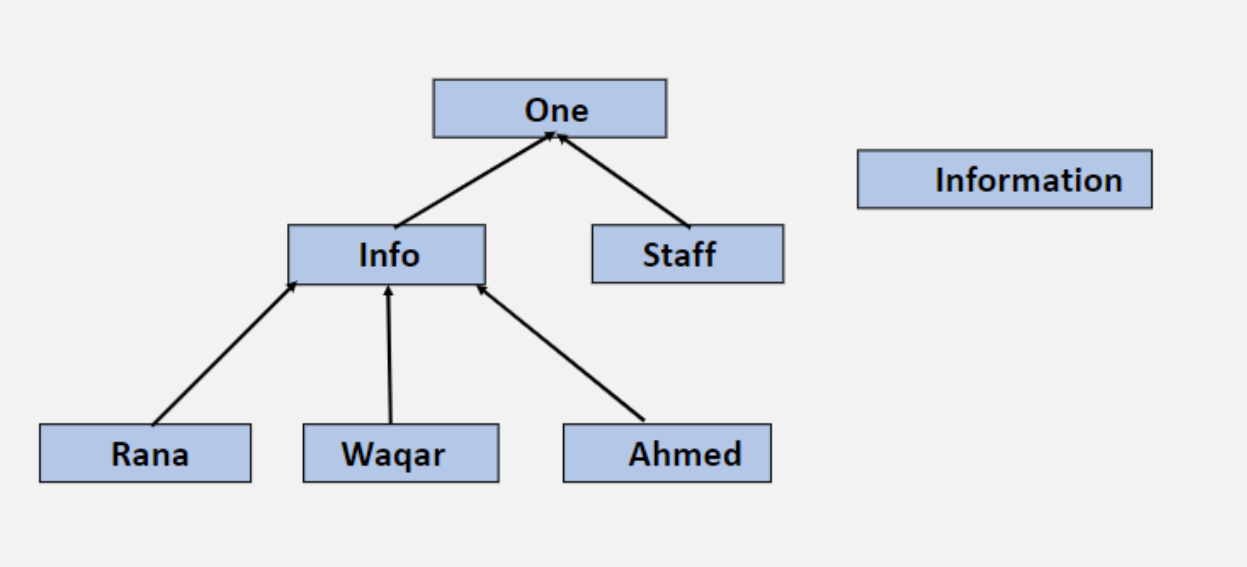
In our project we have used class and object, inheritance, polymorphism, encapsulation, abstraction those are the concepts of object oriented programming.

Class and Object:

In our source code we have used seven different classes such as: One, Info, Rana, Waqar, Ahmed, Staff and Information class. All these class have different objects such as *ptr, s3, s4, s5, a2 and a1 respectively.

Inheritance:

Inheritance is a mechanism in Object Orientated Programming which allows us to derive, acquire, reuse features from one or more classes to one or more classes. The class whose members are being inherited is called the base, parent or mother class, and the class that inherits those members is called derived, child or daughter class.



We have inherited the get() function and show() function in both Info and Staff class from base class One. Again we have inherited data members (name, time, number, age) in derived class Rana, Waqar, Ahmed from Info class.

Polymorphism:

C++ polymorphism means that a call to a member function will cause a different function to be executed depending on the type of object that invokes the function. In our code we have used run time polymorphism which function is pure virtual function like this

```
class One {
        public:
        virtual void get()=0;
        virtual void show()=0;
};
```

In some situations, there are virtual functions which must be overwritten in child classes and have no definition in the base class; those are called pure virtual functions.

We used the get and show function as pure virtual functions so that we can redefine these functions in derived class.

Encapsulation:

Wrapping up data and information under a single unit. Also defined as, binding together the data

and the functions that manipulate them. Encapsulation can lead to data abstraction or data hiding. Encapsulation can be achieved through Access Specifier within a class. For example, sensitive data members are kept private and getters and setter functions are used to access them.

Example:

```
13  class Info:public One
14  {
15    private:
16       string name,time;
17       int num,age;
18    public:
19       string na,ti;
20       int nu,ag;
```

```
38       void setName(string n)
39       {
40          name=n;
41       }
42       string getName()
43       {
44          return name;
45       }
46       void setTime(string t)
47       {
48          time=t;
49       }
50       string getTime()
51       {
52          return time;
53       }
54       void setAge(int a)
55       {
56          age=a;
57       }
58       int getAge()
59       {
60          return age;
61       }
62       void setAppNo(int an)
63       {
```

Name, time, age, num are the private members of Info class to excess and manipulate those variables we have used getter and setter methods inside the Info class. Thus we can say name,

time, num, age and the functions getName(), setName(), getTime(), setTime(), getAge(), setAge() are respectively bonded together. These are the concepts of encapsulation.

Abstract:

Objects cannot be created from classes that contain pure virtual functions. These classes are called abstract. They are classes that can only be used as base classes, and thus are allowed to have pure virtual functions. It can be used to create pointers and take advantage of all its polymorphic abilities.

Example:

main.cpp

```cpp
#include<iostream>
#include<fstream>
#include<string>

using namespace std;

void menu();
class One{
    public:
        virtual void get()=0;
        virtual void show()=0;
};
```

Here One is an abstract class.

Contribution

Sazia Ahmed	• Project idea • Report writing
Anisur Rahman Tajmir	• Project idea • Report writing
Farjahan Akter Boby	• Report writing • Debugging & file operation
Apon Ghosh	• Code writing

Detailed Source Code:

```
#include<iostream>
#include<fstream>
#include<string>

using namespace std;

void menu();
class One{
    public:
        virtual void get()=0;
        virtual void show()=0;
};
class Info:public One
{
  private:
     string name,time;
     int num,age;
  public:
      string na,ti;
      int nu,ag;
      void get()
      {
        system("cls");
        system("color 02");
        cout<<"\n\n\n\n\n\t\t\tEnter the patient name: ";
        cin>>na;
        setName(na);
        cout<<"\n\n\t\t\tEnter age: ";
        cin>>ag;
        setAge(ag);
        cout<<"\n\n\t\t\tEnter the appointment time: ";
        cin>>ti;
        setTime(ti);
        cout<<"\n\n\t\t\tEnter appointment no.: ";
        cin>>nu;
        setAppNo(nu);
      }
      void setName(string n)
      {
          name=n;
      }
      string getName()
```

```
        {
            return name;
        }
        void setTime(string t)
        {
            time=t;
        }
        string getTime()
        {
            return time;
        }
        void setAge(int a)
        {
            age=a;
        }
        int getAge()
        {
            return age;
        }
        void setAppNo(int an)
        {
            num=an;
        }
        int getAppNo()
        {
            return num;
        }

        void show()
        {
          cout<<"\n\n\n\t\t\tPatient's name: "<<name;
          cout<<"\n\n\t\t\tPatient's age: "<<age;
          cout<<"\n\n\t\t\tAppointment no.: "<<num;
          cout<<"\n\n\t\t\tAppointment time: "<<time<<endl;
        }
};

class Rana:public Info
{
  private:
      Info al;
  public:
  void get()
  {
    system("cls");
```

```
    ofstream out("rana.txt",ios::app);
    al.get();
    out<<"Name: "<<al.getName()<<"\tAge: "<<al.getAge()<<"\t\tApp. no.:
"<<al.getAppNo()<<"\t App. time: "<<al.getTime()<<endl;
    out.close();
    cout<<"\n\n\n\t\t\tAppointment has been completed\n\t\t\t";
    system("pause");
    menu();
  }
  void show()
  {
    string line;
    ifstream in("rana.txt");
    system("cls");
    cout<<"\n\n\n\n\t\t\t\t\tDr. Rana's patients:\n";
    cout<<"\t\t-------------------------------------------------------------------
--\n\n";
    if(in.is_open())
    {
       while(in)
        {
            getline(in,line);
            cout<<"\n\n\t\t"<<line<<endl;
        }
      in.close();
      cout<<"\n\n\n\t\t\t";
      system("pause");
      menu();

    }
    else{
      cout<<"\n\n\t\t\tNo data in this file ";
      cout<<"\n\n\t\t\t";
      system("pause");
      menu();
    }
  }
};
class Waqar:public Info
{
    private:
      Info al;
    public:
      void get()
      {
```

```
        system("cls");
        ofstream out("waqar.txt",ios::app);
        al.get();
        out<<"Name: "<<al.getName()<<"\tAge: "<<al.getAge()<<"\t\tApp. no.:
"<<al.getAppNo()<<"\t App. time: "<<al.getTime()<<endl;
        out.close();
        cout<<"\n\n\n\t\t\tAppointment has been completed\n\t\t\t";
        system("pause");
        menu();
      }
      void show()
      {
        string line;
        ifstream in("waqar.txt");
        system("cls");
        cout<<"\n\n\n\n\t\t\t\t\tDr. Waqar's patients:\n";
        cout<<"\t\t-------------------------------------------------------------
------\n\n";
        if(in.is_open())
        {
           while(in)
            {
                getline(in,line);
                cout<<"\n\n\t\t"<<line<<endl;
            }
          in.close();
          cout<<"\n\n\n\t\t\t";
          system("pause");
          menu();

        }
        else{
          cout<<"\n\n\t\t\tNo data in this file ";
          cout<<"\n\n\t\t\t";
          system("pause");
          menu();
        }
      }
};
class Ahmed:public Info
{
    private:
      Info al;
    public:
      void get()
```

```
      {
        system("cls");
        ofstream out("ahmed.txt",ios::app);
        al.get();
        out<<"Name: "<<al.getName()<<"\tAge: "<<al.getAge()<<"\t\tApp. no.:
"<<al.getAppNo()<<"\t App. time: "<<al.getTime()<<endl;
        out.close();
        cout<<"\n\n\n\t\t\tAppointment has been completed\n\t\t\t";
        system("pause");
        menu();
      }
      void show()
      {
        string line;
        ifstream in("ahmed.txt");
        system("cls");
        cout<<"\n\n\n\n\t\t\t\t\tDr. Ahmed's patients:\n";
        cout<<"\t\t-----------------------------------------------------------------
------\n\n";
        if(in.is_open())
        {
           while(in)
            {
                getline(in,line);
                cout<<"\n\n\t\t"<<line<<endl;
            }
          in.close();
          cout<<"\n\n\n\t\t\t";
          system("pause");
          menu();
        }
        else{
          cout<<"\n\n\t\t\tNo data in this file ";
          cout<<"\n\n\t\t\t";
          system("pause");
          menu();
        }
      }
};
class Staff:public One
{
  private:
      string name,age,pos;
      double sal;
  public:
```

```
      void get()
      {
        ofstream out("staff.txt",ios::app);
        {
          system("cls");
          system("color B0");
          cout<<"\n\n\n\n\n\n\n\t\t\tEnter Name = ";
          cin>>name;
          cout<<"\n\t\t\tEnter Age = ";
          cin>>age;
          cout<<"\n\t\t\tEnter Salary = ";
          cin>>sal;
          cout<<"\n\t\t\tEnter Working position = ";
          cin>>pos;
        }
        out<<"\nName: "<<name<<"\tAge: "<<age<<"\t\tSalary: "<<sal<<"\t   Working
position: "<<pos;
        out.close();
        cout<<"\n\n\n\t\t\tYour Information has been saved \n\t\t\t";
        system("pause");
        menu();
      }
      void show()
      {
        char all[999];
        ifstream in("staff.txt");
        if(!in)
        {
          cout<<"\n\n\n\t\t\tFile open error!!!";
          cout<<"\n\n\t\t\t";
          system("pause");
          menu();
        }
        system("cls");
        system("color 71");
        cout<<"\n\n\n\n\t\t\t\t\tStaff informations:\n";
        cout<<"\t\t-------------------------------------------------------------
------";
        while(!(in.eof()))
        {
          in.getline(all,999);
          cout<<"\n\n\t\t"<<all;
        }
        in.close();
        cout<<"\n\n\n\n\t\t\t";
```

```
        system("pause");
        menu();
      }
};
class Information
{
  public:
  void drinfo()
  {
    system("cls");
    system("color F4");
    cout<<"\n\t\t*****************************************\n";
    cout<<"\n\t\t\t (Three Doctor Avaiable) \n\n\t\t [Information and Timing are
given below]\n";
    cout<<"\t\t-----------------------------------------\n";
    cout<<"\t\t\tName: Dr. RANA\n\t\t\t(Skin specialist)\n\n";
    cout<<"\t\t\t[Consulting Hours]:\n\n";
    cout<<"\t\tMonday To Friday\t9AM To 5PM\n";
    cout<<"\t\tSaturday     \t\t9AM To 5PM\n";
    cout<<"\t\tSunday        \t\t OFF\n";
    cout<<"\n\t\t-----------------------------------------\n";
    cout<<"\t\t\tName: Dr. WAQAR\n\t\t\t\t(Child specialist)\n\n";
    cout<<"\t\t\t[Consulting Hours]:\n\n";
    cout<<"\t\tMonday To Friday\t2PM To 10PM\n";
    cout<<"\t\tSaturday     \t\t8AM To 1PM\n";
    cout<<"\t\tSunday        \t\t12PM To 9PM\n";
    cout<<"\n\t\t-----------------------------------------\n";
    cout<<"\t\t\tName: Dr AHMED\n\t\t\t\t(DVM)\n\n";
    cout<<"\t\t\t[Consulting Hours]:\n\n";
    cout<<"\t\tMonday To Friday\t8AM To 5PM\n";
    cout<<"\t\tSaturday     \t\t10AM To 1PM\n";
    cout<<"\t\tSunday        \t\t OFF\n";
    cout<<"\n\t\t*****************************************\n";
    cout<<"\n\n\n\t\t\t";
    system("pause");
    menu();
  }
};
void call_dr()
{
  system("cls");
  system("color 06");
  int choice;
  cout<<"\n\n\n\n\n\n\t\t\t1.\tPress 1 for Dr. Rana \n\n\t\t\t2.\tPress 2 for Dr.
Waqar  \n\n\t\t\t3.\tPress 3 for Dr. Ahmed\n\n\n\t\t\t";
```

```
  cout<<"Please Enter Your Choice : ";
  cin>>choice;
  One *ptr;
  Rana s3;
  Waqar s4;
  Ahmed s5;
  if(choice==1)
  {
    ptr=&s3;
    ptr->get();
  }
  if(choice==2)
  {
    ptr=&s4;
    ptr->get();
  }
  if(choice==3)
  {
    ptr=&s5;
    ptr->get();
  }
  else{
    cout<<"\n\n\n\t\t\tSorry invalid choice.";
    cout<<"\n\t\t\t";
      system("pause");
    menu();
  }
}

void pinfoshow()
{
    system("cls");
    system("color 0D");
    int choice;
    cout<<"\n\n\n\n\n\n\t\t\t1.\tPress 1 for Dr. Rana's patient
\n\n\t\t\t2.\tPress 2 for Dr. Waqar's patient  \n\n\t\t\t3.\tPress 3 for Dr.
Ahmed's patient\n\n\n\t\t\t";
    cout<<"Please Enter Your Choice : ";
    cin>>choice;
    One *ptr;
    Rana s3;
    Waqar s4;
    Ahmed s5;
    if(choice==1)
    {
```

```
        ptr=&s3;
        ptr->show();
    }
    else if(choice==2)
    {
        ptr=&s4;
        ptr->show();
    }
    else if(choice==3)
    {
        ptr=&s5;
        ptr->show();
    }
    else
    {
        cout<<"\n\n\n\t\t\tSorry invalid choice.";
        cout<<"\n\t\t\t";
        system("pause");
        menu();
    }

}
void menu()
{
        system("cls");

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


        cout<<"\t\t\t    |  **********  MAIN MENU   **********  |  \n";
        cout<<"\t\t\t    |      Hospital Management system      |  \n";
        cout<<"\t\t\t    |======================================|  \n";

        cout<<"\n\t----------------------------------------------------------------
------------------\n";

        cout<<"\n\n\t\t1.\tPress 1 For Available Doctor Information\n\n";
        cout<<"\t\t2.\tPress 2 for Doctor Appointment\n\n";
        cout<<"\t\t3.\tPress 3 for Saving Staff Information\n\n";
        cout<<"\t\t4.\tPress 4 for Checking Patient Appointment List\n\n ";
        cout<<"\t\t5.\tPress 5 for Checking Staff Information List\n\n";

        cout<<"\n\t================================================================
==================\n";
        cout<<"\n\n\t\tPlease Enter Your Choice : ";
```

```
        Information a1;
        One *ptr;
        Staff a2;
        int a;
        cin>>a;
        if(a==1)
        {
            a1.drinfo();
        }
        else if(a==2)
        {
            call_dr();
        }
        else if(a==3)
        {
            ptr=&a2;
            ptr->get();

        }
        else if(a==4)
        {
            pinfoshow();
        }
        else if(a==5)
        {
            ptr=&a2;
            ptr->show();
        }
        else
        {
            system("cls");
            cout<<"\n\n\n\n\n\t\t\tInvalid choice!!!\n\n\t\t\tPlease enter
correct option ";
            cout<<"\n\n\n\n\t\t\t";
            system("pause");
            menu();
        }
}

int main()
{
    menu();
    return 0;
}
```