PRIORITY_QUEUE.cpp

ROLL NUMBER : 2002
BATCH : E-10

```
#include <iostream>
#include<stdlib.h>
#include "queues.h"
using namespace std;

int main()
{

        queues q;

        int ch;
        do
        {
                cout<<"\n\n---------Menu-----------\n\n";
                cout<<"\t\t1.Add Patient\n\t\t2.Display List\n\t\t3.Choose patient for Treatment\n\t\t"
                                "4.Exit\n\n\t\t";

                cout<<"Enter Your Choice : ";
                cin>>ch;
                cout<<endl;

                switch(ch)
                {
                        case 1 :        q.enqueue ();
                                                break;
                        case 2 :   q.display();
                                                break;
                        case 3 :   q.dequeue();
                                                break;
                        case 4 :   exit(0);
                                           break;
                        default:   cout<<"Invalid Choice Entered\n\n";
                                                break;
                }
        }
        while(1);

        return 0;
}
```

QUEUES.h

```
/*
 * queues.h
 *
 *  Created on: 08-Jan-2018
 *      Author: e2002
 */

#ifndef QUEUES_H_
#define QUEUES_H_

#include<iostream>
#include<malloc.h>
#include<string>
using namespace std;

struct node
{
	string name;
		string add;
		string ph_no;
		string dis;
	int pri;

	node *link;
};

class queues
{
	node *front,*rear;

	int max=50;

	public:
		queues();
		int isfull();
		int isempty();
		int check(int temp);
		//int priority(int s);
		node *getdata();
		void enqueue();
		void dequeue();
		void display();


};

#endif /* QUEUES_H_ */
```

QUEUES.cpp

```
/*
 * queues.cpp
 *
 *  Created on: 08-Jan-2018
 *      Author: e2002
 */

#include "queues.h"
#include<iostream>
#include<malloc.h>
#include<string>
#include<iomanip>
#include"patient.h"
using namespace std;

queues::queues()
{
	front=rear=NULL;
}

int queues:: isfull()
{
	node *p;
	int count;
	p=front;
	while(p != NULL)
	{
		p=p->link;
		count++;
	}

	if(count==max)
			return 1;
	else
			return 0;

}

int queues:: isempty()
{
	if(rear==NULL)
		return 1;

	else
		return 0;

}

/*int //queues:: priority(int s)
int  priority(int s)
```

```
{
        if(s==1 || s==2)
                return 1;

        else if(s==3 || s==4)
                return 2;

        else
                return 3;
}*/




void queues:: enqueue()
{
        node *n;
        patient pt;
        //n=getdata();
        n=pt.getdata();
        if(isempty())
                front=rear=n;

        else
        {
                rear->link=n;
                rear=n;
        }
}

int queues::check(int temp)
{
        node *p;

        int min=5;

        p=front;

        while(p != NULL)
        {
                if(p->pri < min)
                        min=p->pri;
                p=p->link;
        }
        return min;
}

void queues:: dequeue()
{

        if(isempty())
                {
```

```
        cout<<"No Patient For Treatment\n";
        return;
    }

int min_pri;

min_pri=check(1);
node *current,*previous;

current=front;

if(front->pri==min_pri)
{
    cout<<"Name : "<<front->name<<endl;
    cout<<"Address : "<<front->add<<endl;
    cout<<"Phone Number: "<<front->ph_no<<endl;
    cout<<"Disease : "<<front->dis<<endl;
    cout<<"Priority : "<<front->pri<<endl;

    if(front==rear)
        front=rear=NULL;


    else
        front=front->link;
}

else
{
    while( (current->pri != min_pri) && (current->link != NULL) )
    {
        previous=current;
        current=current->link;
    }

    cout<<"Name : "<<current->name<<endl;
    cout<<"Address : "<<current->add<<endl;
    cout<<"Phone Number: "<<current->ph_no<<endl;
    cout<<"Disease : "<<current->dis<<endl;
    cout<<"Priority : "<<current->pri<<endl;

    if(current==rear)
    {
        rear=previous;
        rear->link=NULL;
        delete current;
    }

    else
    {
        previous->link=current->link;
        delete current;
        }
    }
}

void queues:: display()
{
    if(isempty())
    {
        cout<<"No Patient For Treatment\n";
        return;
    }

    node *p;

    //cout<<left;

    cout<<setw(24)<<"NAME    "
        <<setw(20)<<"ADDRESS "
        <<setw(20)<<"PHONE NO"
        <<setw(20)<<"DISEASE "
        <<setw(20)<<"PRIORITY"<<endl;

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

    p=front;

    while(p != NULL)
    {
        //cout<<left;
        cout<<setw(15)<<"        "<<p->name;
        cout<<setw(15)<<"        "<<p->add;
        cout<<setw(15)<<"        "<<p->ph_no;
        cout<<setw(15)<<"        "<<p->dis;
        cout<<setw(15)<<"        "<<p->pri<<endl<<endl;

        p=p->link;
    }

    cout<<endl;
}
```

PATIENT.h

```
/*
 * patient.h
 *
 *  Created on: 31-Jan-2018
 *      Author: e2002
 */

#ifndef PATIENT_H_
#define PATIENT_H_
#include "queues.h"

class patient
{
	public :
		node * getdata();
};



#endif /* PATIENT_H_ */
```

PATIENT.cpp

```
/*
 * patient.cpp
 *
 *  Created on: 31-Jan-2018
 *      Author: e2002
 */

#include "queues.h"
#include<iostream>
#include<malloc.h>
#include<string>
#include<iomanip>
#include"patient.h"
using namespace std;

int  priority(int s)
{
    if(s==1 || s==2)
        return 1;

    else if(s==3 || s==4)
        return 2;

    else
        return 3;
}

node * patient:: getdata()
//node * queues:: getdata()
{
    node *n;
        n=new node;
        int ch;

        cout<<"Enter Name of Patient : ";
        cin>>n->name;
        cout<<endl;

        cout<<"Enter Address of Patient : ";
        cin>>n->add;
        cout<<endl;

        cout<<"Enter Phone Number of Patient : ";
        cin>>n->ph_no;
        cout<<endl;

        cout<<"Enter Disease of Patient : ";
        cout<<"\n\n\t1.Cancer\n\t2.Accident\n\t3.Blood Pressure\n\t4.Sugar\n\t5.Fever\n\t"
                "6.Cold\n\t";
        cout<<"Enter Option from Above Serviced Disease : ";
        cin>>ch;
        cout<<endl;

        switch (ch)
        {
            case 1 : n->dis="Cancer";
                            break;

            case 2 : n->dis="Accident";
                            break;

            case 3 : n->dis="BP";
                            break;

            case 4 : n->dis="Sugar";
                            break;

            case 5 : n->dis="Fever";
                            break;

            case 6 : n->dis="Cold";
                            break;

            default : cout<<"Choose From Disease Above\n";
                             break;

        }

        n->pri=priority(ch);

        cout<<endl;

        n->link=NULL;

        return n;
}
```