

[Management System]

<< Prepared By:
Raneem Alkhathami
Reema Alzhrani

1
2
3
4
5
6
7
8
9
10
11
12
13
14

{

}

Method:

```
1  bool CheckRecord(int patient_id)
2  {
3      if(head == NULL) {
4          return false;
5      }
6      Patient* temp=head;
7      temp = head;
8      while(temp!=NULL){
9          if(temp->ID==patient_id)
10             temp=temp->next;
11             return true;
12         }
13     return false;
14 }
```

Check Record

```
{
```

```
< Its function is to check before insertion  
whether the record actually exists or not. In  
the event that the record does not exist, the  
record will be added with all the information  
entered. In the event that the record exists,  
a message appears that this ID already  
exists, and there is a record with this ID >
```

```
}
```

Method:

```
void CreatRecord(int ID, string name, int phoneNum, string address,double invoic)
```

```
1  {
2      if(CheckRecord(ID)) {
3          cout<<"There is already a patiant with this record"<<endl;
4          return;
5      } else if(invoic<10) {
6          cout<<"invoic less than 10 will not be added"<<endl;
7          return;
8      }
9      Patient* newNode=new Patient();
10     newNode->ID=ID;
11     newNode->name=name;
12     newNode->phoneNumber=phoneNum;
13     newNode->address=address;
14     newNode->invoic=invoic;
15     newNode->next = NULL;
16     if(head==NULL||head->ID>=newNode->ID){
17         newNode->next=head;
18         head=newNode;
19     }else{
20         Patient* temp=head;
21         while(temp->next!=NULL&&temp->next->ID<newNode->ID){
22             temp=temp->next;
23         }
24         newNode->next = temp->next;
25         temp->next=newNode;
26     }
27     cout<<"Patient record has been Successfully Inserted"<<endl;
28 }
```

Explanation of the methods work

Create Record

```
{
```

```
    < The first step is to call the check Record  
      method to check the ID and if there is a  
      record that holds this ID or not. In the  
      event that the ID is not used, new nodes are  
      added to the patient's record. >
```

```
}
```

```
void SearchRecord(int ID)
```

```
1  {  
2  
3      if(head==NULL) {  
4          cout<<"<<There no pasients with such a record>>"<<endl;  
5          return;  
6      } else{  
7          Patient* temp=head;  
8          while(temp) {  
9              if(temp->ID==ID) {  
10                 cout<<"patient ID number<<"<<temp->ID<<">>"<<endl;  
11                 cout<<"patient name<<"<<temp->name<<">>"<<endl;  
12                 cout<<"patient phone Number<<"<<temp->phoneNumber<<">>"<<endl;  
13                 cout<<"patient address<<"<<temp->address<<">>"<<endl;  
14                 cout<<"patient invoic<<"<<temp->invoic<<">>"<<endl;  
15                 return ;  
16             }  
17             temp= temp->next;  
18         }  
19         if(temp==NULL)  
20             cout<<"<<There no pasients with such a record>>"<<endl;  
21     }  
22 }
```

Search Record

```
{
```

```
< In search, it checks using ID whether the  
patient has a record or not. If it does not  
exist, a message will be printed that the  
patient does not have a record with the  
entered ID, and if it exists, it will search  
for the record with the entered ID and all  
the information entered is printed.>
```

```
}
```

Method:

```
1 void deleteRecord(int ID)
2 {
3
4     if(CheckRecord(ID))
5     {
6         Patient* temp=head;
7         Patient* temp2 = NULL;
8         while(temp!=NULL&&temp->ID!=ID){
9             temp=temp->next;
10        }
11        head=temp->next;
12        free(temp);
13        cout<<"patient record has been deleted"<<endl;
14    }else  cout<<"No patient record was found";
15 }
```


Explanation of the methods work

delete Record

{

< The first step is to make a call to the check record method to check if the entered ID has a record or not. If it does not exist, a message appears. The record to be deleted was not found. In the case of the entered ID present, each node is passed, is it equal to NULL or not, and the entered ID is equal to the ID stored in the node or not.>

}

Method:

```
1 void ShowRecord(int ID)
2 {
3
4     Patient* temp= head;  if(temp == NULL) {
5     cout<<"<<There no pasients with such a record>>"<<endl;
6     } else{
7     cout<<"Patient ID\tPatient name\tPatient phone Number " <<"\tPatient address\tPatient
8     invoic"<<endl;
9     Patient* temp= head;
10    while(temp!=NULL)
11    {
12        cout<<temp->ID<<"\t\t"<<temp->name<<"\t\t" <<temp->phoneNumber<<"\t\t"
13        <<temp->address<<"\t\t"<<temp->invoic<<"\t\t" <<endl;
14        temp=temp->next;
15    }
16 }
```

Explanation of the methods work

Show Record

```
{
```

```
< it is function is to show all records that  
have been recorded with all the information  
and is also used in the case of using other  
methods such as deletion and update to make a  
show after applying the methods>
```

```
}
```

Method:

```
1 void updateRecord(int ID)
2 {
3     Patient* temp=head;
4     if(CheckRecord(ID)) {
5         while (temp != NULL)
6         {
7             if (temp->invoic > 100)
8             {
9                 temp->invoic = ( temp->invoic * 0.15 ) + temp->invoic;
10                cout << "Invoice Updated\n";
11                return;
12            } else if (temp->invoic < 10) {
13                temp->invoic;
14                cout << "Invoice Updated\n";
15            }
16            temp = temp->next;
17        }
18    } else
19        cout << "There is no patients by this ID to update their invoice\n";
20 }
```

Explanation of the methods work

update Record

{

< it is function is to show all records that
have been recorded with all the information
and is also used in the case of using other
methods such as deletion and update to make a
show after applying the methods>

}

1
2
3
4
5
6
7
8
9
10
11
12
13
14

{
|
}
}

OUTPUT;

