

<b>Ex. No. 6</b>	<b>Linked list Implementation of Stack and Queue</b>
<b>Date of Exercise</b>	23-08-2017

## Aim:

To simulate the working of a queue and stack of integers using a linked list.

## Algorithm:

1. Start.
2. For stack operations,
3. Create struct node, int info and node\*link
4. Node\*front=null; node\*rear= null; node\*temp;
5. Create the push function with necessary steps.
6. Create the pop function with necessary steps.
7. Create the display function with necessary steps.
8. Inside the main function we are having switch case for 4 inputs
9. And the corresponding functions are called
10. For queue operations
11. Create struct node, int info and node\*link
12. Node\*top=null; node\*temp;
13. Create the add queue function with necessary steps.
14. Create the delete queue function with necessary steps.
15. Create the display function with necessary steps.
16. Inside the main function we are having switch case for 4 inputs.
17. And the corresponding functions are called.
18. End

## Source Code:

```
/* Linked list implementation of stack and queue*/
#include <iostream>
using namespace std;
struct node
{
    int info;
    node*link;
};
node*FRONT=NULL;
node*REAR=NULL;
node*top=NULL;
```

```
node*temp;

//Stack operations
void push(int data)
{
    temp=new node;
    temp->info=data;
    temp->link=top;
    top=temp;
}

int pop()
{
    if (top==NULL)
    {
        cout<<"Stack is empty\n";
        cout<<endl;
    }
    else
    {
        int data;
        temp=top;
        data=top->info;
        top=top->link;
        delete(temp);
        cout<<"Deleted element is: "<<data<<endl<<endl;
    }
}

void display1()
{
    node*monk;
    monk=top;
    cout<<"\n The elements in the stack are: ";
    while (monk!=NULL)
    {
        cout<<monk->info<<"-";
        monk =monk->link;
    }
    cout<<"NULL"<<endl<<endl;
}

// Queue operations
void addque(int data)
{
    temp=new node;
```

```
if(temp==NULL)
{
    cout<<"\n Queue is full";
    return;
}
else
{
    temp->info=data;
    temp->link=NULL;
}
if(FRONT==NULL)
{
    FRONT=REAR=temp;
    return;
}
else
{
    REAR->link=temp;
    REAR=temp;
    return;
}
}
int delq()
{
    if (FRONT==NULL)
    {
        cout<<"\nQueue is empty\n"<<endl;
    }
    else
    {
        int data;
        temp=FRONT;
        data=FRONT->info;
        FRONT=FRONT->link;
        cout<<"\nThe deleted element is: "<<data<<endl<<endl;
        delete(temp);
        return data;
    }
}
void display()
{
    node*monk;
    monk=FRONT;
    cout<<"\nThe elements in the queue are: ";
    while (monk!=NULL)
    {
```

```
        cout<<monk->info<<"->";
        monk =monk->link;
    }
    cout<<"NULL\n"<<endl;
}

int main()
{
    int op;
    cout<<"\n Choices to do.....\n";
    cout<<"\n 1.Stack operations";
    cout<<"\n 2.Queue operations";
    cout<<"\n 3.Exit";
    cout<<"\n\n Enter your choice: ";
    cin>>op;
    cout<<"\n";
    switch(op)
    {
    case 1:
        int c;
        while(1)
        {
            cout<<"Functions to perform:\n1.PUSH\n2.POP\n3.DISPLAY\n4.EXIT\n";
            cout<<"\nEnter your choice: ";
            cin>>c;
            switch(c)
            {
            case 1:
                cout<<"Enter a number: ";
                int n;
                cin>>n;
                cout<<"\n";
                push(n);
                break;
            case 2:
                pop();
                break;

            case 3:
                display1();
                break;
            case 4:
                return 0;
                break;
            }
        }
    }
}
```

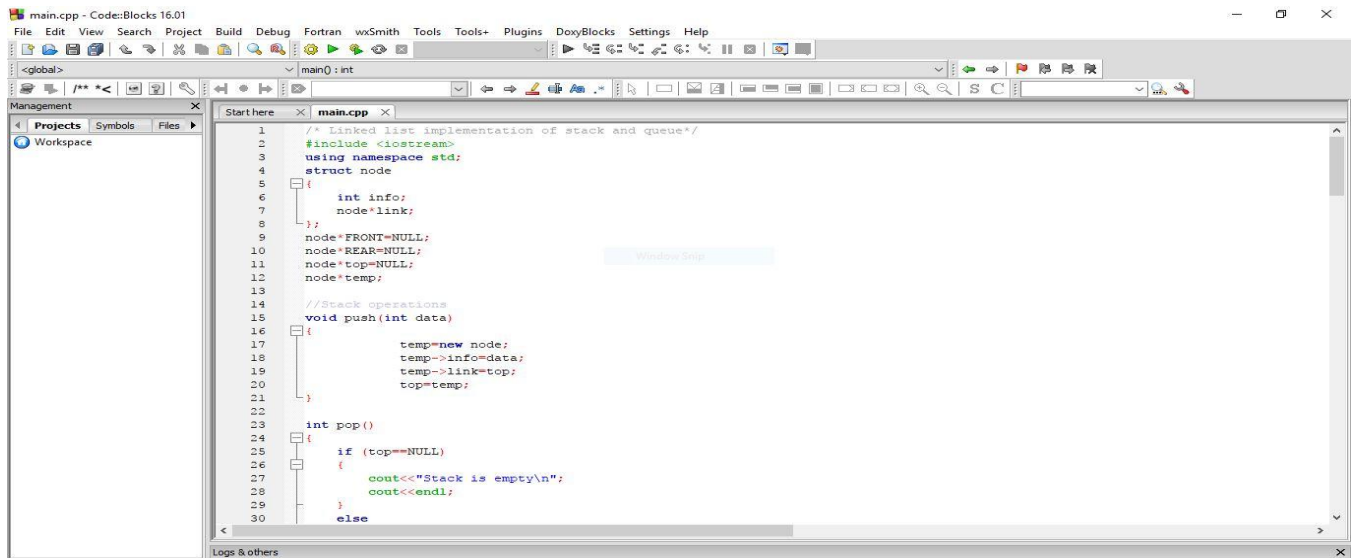
```
case 2:
    int opt;
    while(1)
    {
        cout<<"Functions to
perform:\n1.ENQUEUE\n2.DEQUEUE\n3.DISPLAY\n4.EXIT\n";
        cout<<"\nEnter your choice:";
        cin>>opt;
        switch(opt)
        {
            case 1:
                cout<<"\nEnter a number: ";
                int n;
                cin>>n;
                cout<<"\n";
                addque(n);
                break;

            case 2:
                delq();
                break;

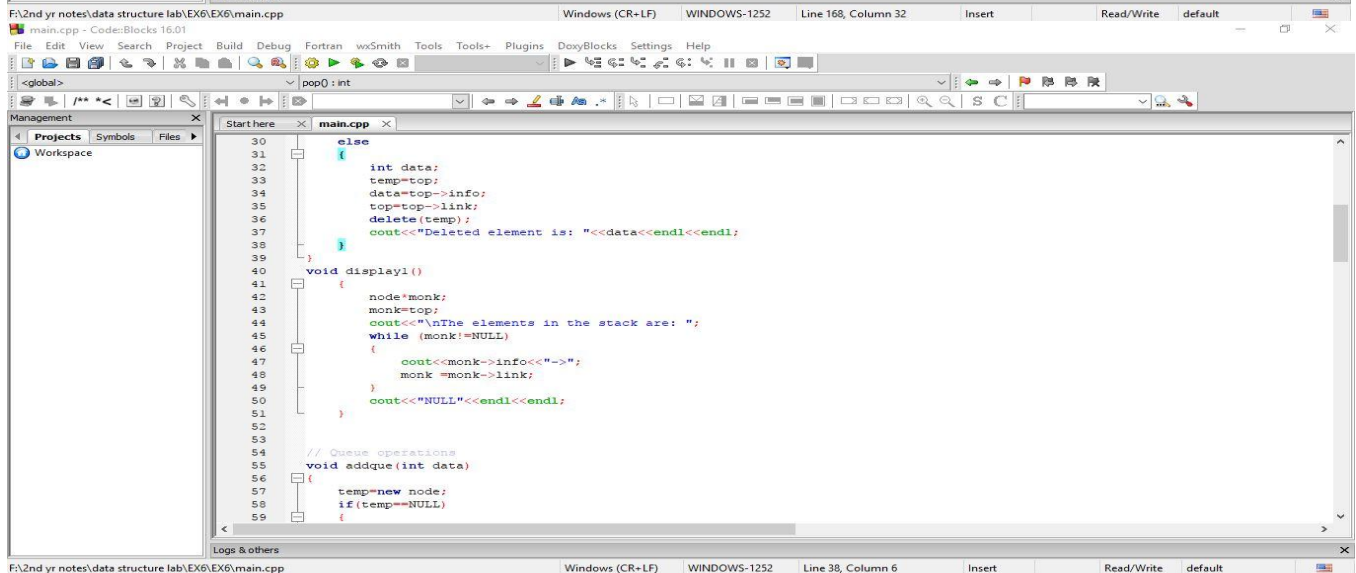
            case 3:
                display();
                break;
            case 4:
                return 0;
                break;
        }
    }
case 3:
    return 0;
}
```

# Object Oriented Programming in C++

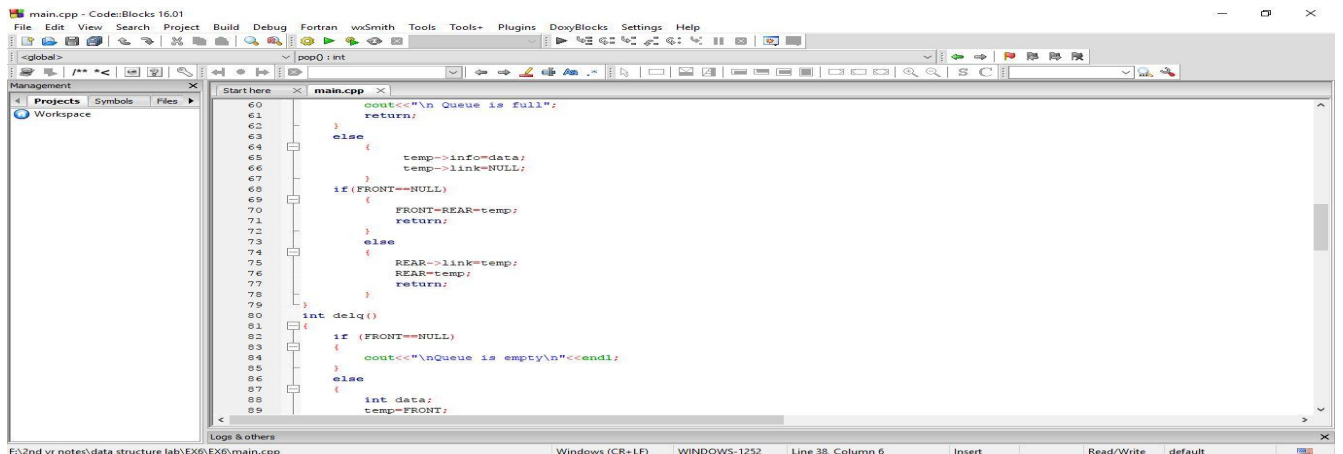
## Output:



```
1  /* Linked list implementation of stack and queue */
2  #include <iostream>
3  using namespace std;
4  struct node
5  {
6      int info;
7      node* link;
8  };
9  node* FRONT=NULL;
10 node* REAR=NULL;
11 node* top=NULL;
12 node* temp;
13
14 //Stack operations
15 void push(int data)
16 {
17     temp=new node;
18     temp->info=data;
19     temp->link=top;
20     top=temp;
21 }
22
23 int pop()
24 {
25     if (top==NULL)
26     {
27         cout<<"Stack is empty\n";
28         cout<<endl;
29     }
30     else
```



```
31     {
32         int data;
33         temp=top;
34         data=temp->info;
35         top=temp->link;
36         delete(temp);
37         cout<<"Deleted element is: "<<data<<endl<<endl;
38     }
39 }
40
41 void display()
42 {
43     node* monk;
44     monk=top;
45     cout<<"\nThe elements in the stack are: ";
46     while (monk!=NULL)
47     {
48         cout<<monk->info<<"->";
49         monk =monk->link;
50     }
51     cout<<"NULL"<<endl<<endl;
52 }
53
54 // Queue operations
55 void addque(int data)
56 {
57     temp=new node;
58     if (temp==NULL)
59     {
```



```
60     cout<<"\n Queue is full";
61     return;
62 }
63 else
64 {
65     temp->info=data;
66     temp->link=NULL;
67 }
68 if (FRONT==NULL)
69 {
70     FRONT=REAR=temp;
71     return;
72 }
73 else
74 {
75     REAR->link=temp;
76     REAR=temp;
77     return;
78 }
79
80 int delq()
81 {
82     if (FRONT==NULL)
83     {
84         cout<<"\nQueue is empty\n"<<endl;
85     }
86     else
87     {
88         int data;
89         temp=FRONT;
```

# Object Oriented Programming in C++

The image displays three sequential screenshots of a C++ IDE (Code::Blocks 16.01) showing the implementation of a linked list with stack and queue operations. The code is written in `main.cpp` and includes a `pop0` function.

**First Screenshot (Lines 90-119):** Shows the initial setup of the linked list. It includes a `display()` function that traverses the list and prints the elements. The `main()` function prompts the user for a choice to perform operations.

```
90 data=FRONT->info;
91 FRONT->link;
92 cout<<"\nThe deleted element is: "<<data<<endl<<endl;
93 delete (temp);
94 return data;
95 }
96 }
97 void display()
98 {
99     node*monk;
100     monk=FRONT;
101     cout<<"\nThe elements in the queue are: ";
102     while (monk!=NULL)
103     {
104         cout<<monk->info<<"->";
105         monk =monk->link;
106     }
107     cout<<"NULL\n"<<endl;
108 }
109
110 int main()
111 {
112     int op;
113     cout<<"\n Choices to do....\n";
114     cout<<"\n 1.Stack operations";
115     cout<<"\n 2.Queue operations";
116     cout<<"\n 3.Exit";
117     cin>>op;
118     cin>>op;
119     cout<<"\n";
```

**Second Screenshot (Lines 120-149):** Shows the implementation of the `switch` statement for the first set of operations. It includes a `while` loop for the first case and a `switch` statement for the second case.

```
120 switch(op)
121 {
122     case 1:
123         int c;
124         while(1)
125         {
126             cout<<"Functions to perform:\n1.PUSH\n2.POP\n3.DISPLAY\n4.EXIT\n";
127             cout<<"\nEnter your choice: ";
128             cin>>c;
129             switch(c)
130             {
131                 case 1:
132                     cout<<"Enter a number: ";
133                     int n;
134                     cin>>n;
135                     cout<<"\n";
136                     push(n);
137                     break;
138                 case 2:
139                     pop();
140                     break;
141                 case 3:
142                     display();
143                     break;
144                 case 4:
145                     return 0;
146                     break;
147             }
148         }
149 }
```

**Third Screenshot (Lines 150-179):** Shows the implementation of the `switch` statement for the second set of operations. It includes a `while` loop for the first case and a `switch` statement for the second case.

```
150 case 2:
151     int opt;
152     while(1)
153     {
154         cout<<"Functions to perform:\n1.ENQUEUE\n2.DEQUEUE\n3.DISPLAY\n4.EXIT\n";
155         cout<<"\nEnter your choice:";
156         cin>>opt;
157         switch(opt)
158         {
159             case 1:
160                 cout<<"\nEnter a number: ";
161                 int n;
162                 cin>>n;
163                 cout<<"\n";
164                 addque(n);
165                 break;
166             case 2:
167                 delq();
168                 break;
169             case 3:
170                 display();
171                 break;
172             case 4:
173                 return 0;
174                 break;
175             case 3:
176                 break;
177         }
178     }
179 }
```

# Object Oriented Programming in C++

■ "F:\2nd yr notes\data structure lab\EX6\EX6\main.exe"

Choices to do.....

- 1.Stack operations
- 2.Queue operations
- 3.Exit

Enter your choice: 1

Functions to perform:

- 1.PUSH
- 2.POP
- 3.DISPLAY
- 4.EXIT

Enter your choice: 1

Enter a number: 123

Functions to perform:

- 1.PUSH
- 2.POP
- 3.DISPLAY
- 4.EXIT

Enter your choice: 1

Enter a number: 124

Functions to perform:

- 1.PUSH
- 2.POP
- 3.DISPLAY
- 4.EXIT

Enter your choice: 3

The elements in the stack are: 124->123->NULL

■ Select "F:\2nd yr notes\data structure lab\EX6\EX6\main.exe"

Functions to perform:

- 1.PUSH
- 2.POP
- 3.DISPLAY
- 4.EXIT

Enter your choice: 3

The elements in the stack are: 124->123->NULL

Functions to perform:

- 1.PUSH
- 2.POP
- 3.DISPLAY
- 4.EXIT

Enter your choice: 2

Deleted element is: 124

Functions to perform:

- 1.PUSH
- 2.POP
- 3.DISPLAY
- 4.EXIT

Enter your choice: 2

Deleted element is: 123

Functions to perform:

- 1.PUSH
- 2.POP
- 3.DISPLAY
- 4.EXIT

Enter your choice: 2

Stack is empty

■ "F:\2nd yr notes\data structure lab\EX6\EX6\main.exe"

Choices to do.....

- 1.Stack operations
- 2.Queue operations
- 3.Exit

Enter your choice: 2

Functions to perform:

- 1.ENQUEUE
- 2.DEQUEUE
- 3.DISPLAY
- 4.EXIT

Enter your choice:1

Enter a number: 123

Functions to perform:

- 1.ENQUEUE
- 2.DEQUEUE
- 3.DISPLAY
- 4.EXIT

Enter your choice:1

Enter a number: 234

Functions to perform:

- 1.ENQUEUE
- 2.DEQUEUE
- 3.DISPLAY
- 4.EXIT

Enter your choice:3

The elements in the queue are: 123->234->NULL



# Object Oriented Programming in C++

```
"F:\2nd yr notes\data structure lab\EX6\EX6\main.exe"
3.DISPLAY
4.EXIT

Enter your choice:3

The elements in the queue are: 123->234->NULL

Functions to perform:
1.ENQUEUE
2.DEQUEUE
3.DISPLAY
4.EXIT

Enter your choice:2

The deleted element is: 123

Functions to perform:
1.ENQUEUE
2.DEQUEUE
3.DISPLAY
4.EXIT

Enter your choice:2

The deleted element is: 234

Functions to perform:
1.ENQUEUE
2.DEQUEUE
3.DISPLAY
4.EXIT

Enter your choice:2

Queue is empty
```

## Video URL:

<https://youtu.be/KjENb7sTBL4>

## Result:

The program of linked implementation of stack and queue is implemented successfully and the output is verified.