

```
1  #include <iostream>
2
3  using namespace std;
4  struct Node
5  {
6      int data;
7      Node* next;
8  };
9  class Queue
10 {
11 private:
12     Node* front;
13     Node* rear;
14     int count;
15 public:
16     Queue()
17     {
18         front = NULL;
19         rear = NULL;
20         count = 0;
21     }
22     bool isEmpty(); // check if Queue is empty
23     void enqueue(int); // add new element in last position of Queue
24     void dequeue(); // remove last element
25     void peek(); // Show last element
26     void displayCount(); // show number of elements in Queue
27 };
28 bool Queue::isEmpty()
29 {
30     return rear == NULL;
31 }
32 void Queue::enqueue(int val)
33 {
34     Node* newNode = new Node;
35     newNode ->data = val;
36     newNode ->next = NULL;
37     if (isEmpty())
38     {
39         front = rear = newNode;
40         count++;
41     }
42     else
43     {
44         rear ->next = newNode;
45         rear = newNode;
46         count++;
47     }
48 }
49 void Queue::dequeue()
50 {
51     if (front == NULL)
52     {
53         cout << "Queue is empty" << endl;
```

```
54     }
55     else
56     {
57         Node* temp = front;
58         int data = temp->data;
59         // if there is only one element in queue then remove it
60         if (front == rear)
61         {
62             front = rear = NULL;
63             delete(temp);
64         }
65         else
66         {
67             //if there are more than one elements
68             front = front->next;
69             delete(temp);
70         }
71         cout << "\nDeleted :" << data << endl;
72         count--;
73     }
74 }
75
76 void Queue::displayCount()
77 {
78     cout << "\nElements in Queue: " << count << endl;
79 }
80 void Queue::peek()
81 {
82     if (!isEmpty())
83     {
84         cout << "\nFirst Element :" << front ->data << endl;
85     }
86 }
87
88 void main()
89 {
90     Queue q;
91     q.dequeue();
92     q.enqueue(10);
93     q.enqueue(20);
94     q.enqueue(30);
95     q.enqueue(40);
96     q.peek();
97     q.enqueue(50);
98     q.displayCount();
99     q.dequeue();
100    q.dequeue();
101    q.dequeue();
102    q.dequeue();
103    q.dequeue();
104    q.dequeue();
105    q.displayCount();
106
```

```
107     getchar();  
108 }
```