

## Question01.cpp

```
1 // <----Lab 06 - Queues---->
2
3 // Q1. Write the definition of all the functions listed in Example 01.
4
5 #include <iostream>
6 using namespace std;
7
8 class Queue
9 {
10 private:
11     int front, rear;
12     unsigned capacity;
13     int *array;
14
15 public:
16     Queue(int capacity)
17     {
18         array = new int[capacity];
19         front = -1;
20         rear = -1;
21         this->capacity = capacity;
22     }
23
24     bool isEmpty()
25     {
26         return (rear == -1 && front == -1);
27     }
28
29     bool isFull()
30     {
31         return ((rear + 1) % capacity == front);
32     }
33
34     void enqueue(int element)
35     {
36         if (isFull())
37         {
38             cerr << "Queue is full cannot enqueue." << endl;
39             return;
40         }
41         else if (isEmpty())
42         {
43             front = rear = 0;
44         }
45         else
46         {
47             rear = (rear + 1) % capacity;
48         }
49         array[rear] = element;
50         cout << array[rear] << " is enqueued in the queue." << endl;
51     }
52
53     void dequeue()
```

```
54     {
55         if (isEmpty())
56         {
57             cerr << "Queue is Empty cannot dequeue." << endl;
58             return;
59         }
60         else if (front == rear)
61         {
62             cout << array[front] << " is dequeued from the queue." << endl;
63             rear = front = -1;
64         }
65         else
66         {
67             cout << array[front] << " is dequeued from the queue." << endl;
68             front = (front + 1) % capacity;
69         }
70     }
71
72     int peek()
73     {
74         if (isEmpty())
75         {
76             cerr << "Queue is empty." << endl;
77         }
78         return array[front];
79     }
80
81     int queue_size()
82     {
83         if (isEmpty())
84             return 0;
85         else if (rear >= front)
86         {
87             return rear - front + 1;
88         }
89         else
90         {
91             return (capacity - front + rear) % capacity;
92         }
93     }
94 };
95
96 int main()
97 {
98     Queue q(10);
99     q.enqueue(1);
100    q.enqueue(2);
101    q.enqueue(3);
102    q.enqueue(4);
103    q.enqueue(5);
104    q.enqueue(6);
105    q.enqueue(7);
106    q.enqueue(8);
107    cout << "Front element " << q.peek() << endl;
108    cout << "Size of queue " << q.queue_size() << endl;
109    return 0;
```

```
110 }  
111
```