Question3.cpp

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

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

```
110     q.dequeue();
111     cout << "Front element " << q.peek() << endl;
112     cout << "Size of queue " << q.queue_size() << endl;
113     return 0;
114  }
115</pre>
```