12/12/23, 11:46 PM Queue.h

Queue.h

```
#ifndef QUEUE_H
 2
    #define QUEUE_H
 3
 4
   #include <iostream>
 5
    using namespace std;
6
7
   template <class T>
    class Node
8
9
   {
    public:
10
        T data;
11
        Node *next;
12
13
        Node(T data)
14
15
            this->data = data;
16
            next = nullptr;
17
18
    };
19
20 template <class T>
21
    class Queue
22
   {
23
   private:
24
        Node<T> *rear;
25
        Node<T> *front;
26
        int size;
27
        int capacity;
28
29
    public:
        Queue(int capacity) : rear(nullptr), front(nullptr)
30
31
32
            this->capacity = capacity;
33
            size = 0;
34
35
36
        bool isEmpty()
37
            return (rear == nullptr && front == nullptr);
38
39
        }
40
41
        bool isFull()
42
43
            return (size == capacity);
44
        }
45
        void Put(T value)
46
47
            Node<T> *newnode = new Node<T>(value);
48
49
            if (isFull())
50
                cerr << "Cannot enqueue queue is full" << endl;</pre>
51
52
                delete newnode;
53
                return;
```

```
54
55
            else if (isEmpty())
56
57
                rear = front = newnode;
58
            }
59
            else
60
            {
                rear->next = newnode;
61
62
                rear = newnode;
63
                ++size;
64
65
        }
66
67
        void dequeue()
68
        {
69
            if (isEmpty())
70
                 cerr << "Queue is empty cannot dequeu." << endl;</pre>
71
72
                 return;
73
74
            else if (rear == front)
75
                rear = front = nullptr;
76
77
78
            else
79
                Node<T> *temp = front;
80
81
                front = front->next;
82
                delete temp;
83
84
            --size;
85
        }
86
        T Get()
87
88
            T value = front->data;
89
90
            dequeue();
            return value;
91
92
        }
93
   };
94 #endif
```