

Queue.h

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

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