

```

1  #include "cnPtrQueue.h"
2  #include <cassert>
3  using namespace std;
4
5  namespace CS3358_FA2019_A5P2
6  {
7      // to be implemented (part of assignment)
8      cnPtrQueue::cnPtrQueue() : numItems(0){}
9
10     bool cnPtrQueue::empty() const
11     {
12         return (outStack.empty() && inStack.empty());
13     }
14
15     cnPtrQueue::size_type cnPtrQueue::size() const
16     {
17         return numItems;
18     }
19
20     CNode* cnPtrQueue::front()
21     {
22         // Checking precondition (not empty)
23         assert(!inStack.empty() || !outStack.empty());
24         if(outStack.empty())
25         {
26             while(!inStack.empty())
27             {
28                 outStack.push(inStack.top());
29                 inStack.pop();
30             }
31         }
32         return outStack.top();
33     }
34
35     void cnPtrQueue::push(CNode* cnPtr)
36     {
37         inStack.push(cnPtr);
38         ++numItems;
39     }
40
41     void cnPtrQueue::pop()
42     {
43         // Checking precondition (not empty)
44         assert(!inStack.empty() || !outStack.empty());
45         if(outStack.empty())
46         {
47             while(!inStack.empty())
48             {
49                 outStack.push(inStack.top());
50                 inStack.pop();
51             }
52         }
53         outStack.pop();
54         --numItems;
55     }
56
57 }
58

```