```cpp
#include "cnPtrQueue.h"
#include <cassert>
using namespace std;

namespace CS3358_FA2019_A5P2
{
   // to be implemented (part of assignment)
   cnPtrQueue::cnPtrQueue() : numItems(0){}

   bool cnPtrQueue::empty() const
   {
      return (outStack.empty() && inStack.empty());
   }

   cnPtrQueue::size_type cnPtrQueue::size() const
   {
      return numItems;
   }

   CNode* cnPtrQueue::front()
   {
      // Checking precondition (not empty)
      assert(!inStack.empty() || !outStack.empty());
      if(outStack.empty())
      {
         while(!inStack.empty())
         {
            outStack.push(inStack.top());
            inStack.pop();
         }
      }
      return outStack.top();
   }

   void cnPtrQueue::push(CNode* cnPtr)
   {
      inStack.push(cnPtr);
      ++numItems;
   }

   void cnPtrQueue::pop()
   {
      // Checking precondition (not empty)
      assert(!inStack.empty() || !outStack.empty());
      if(outStack.empty())
      {
         while(!inStack.empty())
         {
            outStack.push(inStack.top());
            inStack.pop();
         }
      }
      outStack.pop();
      --numItems;
   }

}

```