```cpp
#include <iostream>
using namespace std;

template <class T>
class Queue
{
private:
    T* queueArray;
    int capacity;
    int numItems;
public:
    Queue(int);
    ~Queue();
    void enqueue(T);
    void dequeue(T&);
    bool isEmpty() const;
    bool isFull() const;
    void display() const;
};
template <class T>
Queue<T>::Queue(int s)
{
    queueArray = new T[s];
    capacity = s;
    numItems = 0;
}
template <class T>
Queue<T>::~Queue()
{
    delete[] queueArray;
}
template <class T>
void Queue<T>::enqueue(T num)
{
    if (isFull())
        cout << "The queue is full.\n";
    else
    {
        queueArray[numItems] = num;
        numItems++;
    }
}
template <class T>
void Queue<T>::dequeue(T& num)
{
    if (isEmpty())
        cout << "The queue is empty.\n";
    else
    {
        num = queueArray[0];
        for (int i = 0; i < numItems - 1; i++)
            queueArray[i] = queueArray[i + 1];
        numItems--;
    }
}
template <class T>
bool Queue<T>::isEmpty() const
{
    return numItems == 0;
}
template <class T>
bool Queue<T>::isFull() const
{
    return numItems == capacity;
}
template <class T>
void Queue<T>::display() const
{
    if (isEmpty())
    {
        cout << "The queue is empty.\n";
        return;
    }
    cout << "\nThese are elements of the Queue.\n";
    for (int i = 0; i < numItems; i++)
    {
        cout << queueArray[i] << endl;
    }
}
int main()
{
    int catchVar;
    string strCatchVar;
    Queue<int> iQueue(5);

    iQueue.enqueue(5);
    iQueue.display();
    iQueue.enqueue(12);
    iQueue.display();
    iQueue.enqueue(8);
    iQueue.display();
    iQueue.enqueue(15);
    iQueue.display();
    iQueue.enqueue(17);
    iQueue.display();
    iQueue.enqueue(5);
    iQueue.display();

    iQueue.dequeue(catchVar);
    iQueue.display();
    iQueue.dequeue(catchVar);
    iQueue.display();
    iQueue.enqueue(99);
    iQueue.display();

    Queue<string> sQueue(5);
    sQueue.enqueue("Jack");
    sQueue.display();
    sQueue.enqueue("Jeff");
    sQueue.display();
    sQueue.enqueue("Joe");
    sQueue.display();
    sQueue.enqueue("John");
    sQueue.display();

    sQueue.dequeue(strCatchVar);
    sQueue.display();
    sQueue.dequeue(strCatchVar);
    sQueue.display();
    sQueue.dequeue(strCatchVar);
    sQueue.display();

    return 0;
}

```