```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
    {
```

```
50          num = queueArray[0];
51          for (int i = 0; i < numItems - 1; i++)
52              queueArray[i] = queueArray[i + 1];
53          numItems--;
54      }
55  }
56  template <class T>
57  bool Queue<T>::isEmpty() const
58  {
59      return numItems == 0;
60  }
61  template <class T>
62  bool Queue<T>::isFull() const
63  {
64      return numItems == capacity;
65  }
66  template <class T>
67  void Queue<T>::display() const
68  {
69      if (isEmpty())
70      {
71          cout << "The queue is empty.\n";
72          return;
73      }
74      cout << "\nThese are elements of the Queue.\n";
75      for (int i = 0; i < numItems; i++)
76      {
77          cout << queueArray[i] << endl;
78      }
79  }
80  int main()
81  {
82      int catchVar;
83      string strCatchVar;
84      Queue<int> iQueue(5);
85
86      iQueue.enqueue(5);
87      iQueue.display();
88      iQueue.enqueue(12);
89      iQueue.display();
90      iQueue.enqueue(8);
91      iQueue.display();
92      iQueue.enqueue(15);
93      iQueue.display();
94      iQueue.enqueue(17);
95      iQueue.display();
96      iQueue.enqueue(5);
97      iQueue.display();
98
```

```
 99      iQueue.dequeue(catchVar);
100      iQueue.display();
101      iQueue.dequeue(catchVar);
102      iQueue.display();
103      iQueue.enqueue(99);
104      iQueue.display();
105
106      Queue<string> sQueue(5);
107      sQueue.enqueue("Jack");
108      sQueue.display();
109      sQueue.enqueue("Jeff");
110      sQueue.display();
111      sQueue.enqueue("Joe");
112      sQueue.display();
113      sQueue.enqueue("John");
114      sQueue.display();
115
116      sQueue.dequeue(strCatchVar);
117      sQueue.display();
118      sQueue.dequeue(strCatchVar);
119      sQueue.display();
120      sQueue.dequeue(strCatchVar);
121      sQueue.display();
122
123      return 0;
124  }
125
```