

Queue Template*Last Updated: 12 August 2016*

```
template <class T>
class Queue1
    ///! is modeled by string of T
    ///!   exemplar self
{
public: // Standard Operations
    Queue1 ();
        ///! replaces self
        ///! ensures: self = <>
    ~Queue1 ();
    void clear (void);
        ///! clears self
    void transferFrom (Queue1& source);
        ///! replaces self
        ///! clears source
        ///! ensures: self = #source
    Queue1& operator = (Queue1& rhs);
        ///! replaces self
        ///! restores rhs
        ///! ensures: self = rhs

    // Queue1 Specific Operations
    void enqueue (T& x);
        ///! updates self
        ///! clears x
        ///! ensures: self = #self * <#x>
    void dequeue (T& x);
        ///! updates self
        ///! replaces x
        ///! requires: self /= <>
        ///! ensures: <x> is prefix of #self  and  self = #self[1, |#self|)
    void replaceFront (T& x);
        ///! updates self, x
        ///! requires: self /= <>
        ///! ensures: <x> is prefix of #self  and  self = <#x> * #self[1, |#self|)
    T& front (void);
        ///! restores self
        ///! requires: self /= <>
        ///! ensures: <front> is prefix of self
    Integer length (void);
        ///! restores self
        ///! ensures: length = |self|

private: // Representation
}
```