

```
#ifndef QUEUE_H_
#define QUEUE_H_

/**
 * The Queue is a pure virtual (abstract) interface of a queue. Queue
 * implementations should inherit from and implement this interface.
 */
template <typename T>
class Queue {
public:
    virtual ~Queue() { }; // An empty virtual destructor.
    virtual void enqueue(T item) = 0; // Put an item on the end of the queue.
    virtual T dequeue() = 0; // Remove and return front item.
    virtual int getSize() = 0; // Returns current number of items.
    virtual bool isEmpty() = 0; // Returns true if the queue is empty.
    virtual T getFront() = 0; // Returns (but does not remove) front.
};

#endif // QUEUE_H_
```