

ADT handout:

An ADT (abstract data type) consists of 2 things

- a set S of 'structures' called states.
- an associated set of operations on states.
 - Access functions: return info. about a state, don't change it
 - Manipulation Procedures: change state, return nothing.

Each instance of an ADT has a unique history of states.

EX. Integer Queue

• states: finite sequences of integers.

• Operations:

- Manip. Proc: Enqueue()
Dequeue()

- Access funcs: getFront()
getLength()
isEmpty()

<u>Op</u>	<u>state</u>	<u>Val</u>
-	()	-
Enqueue(5)	(5)	-
Enqueue(1)	(5, 1)	-
Enqueue(7)	(5, 1, 7)	-
Dequeue()	(1, 7)	-
getLength()	(1, 7)	2
getFront()	(1, 7)	1
isEmpty()	(1, 7)	false

What do Dequeue() and getFront() do when isEmpty() is true.

14

We define precondition for each
OP. that must be
true when the OP. is called.

Ex. Precond. for both `Dequeue()`
and `getFront()` is `not isEmpty()`.

what do we do if a Precond. is
Violated. ?

in C: print an error msg.
kill Prog.

in C++: throw an exception.

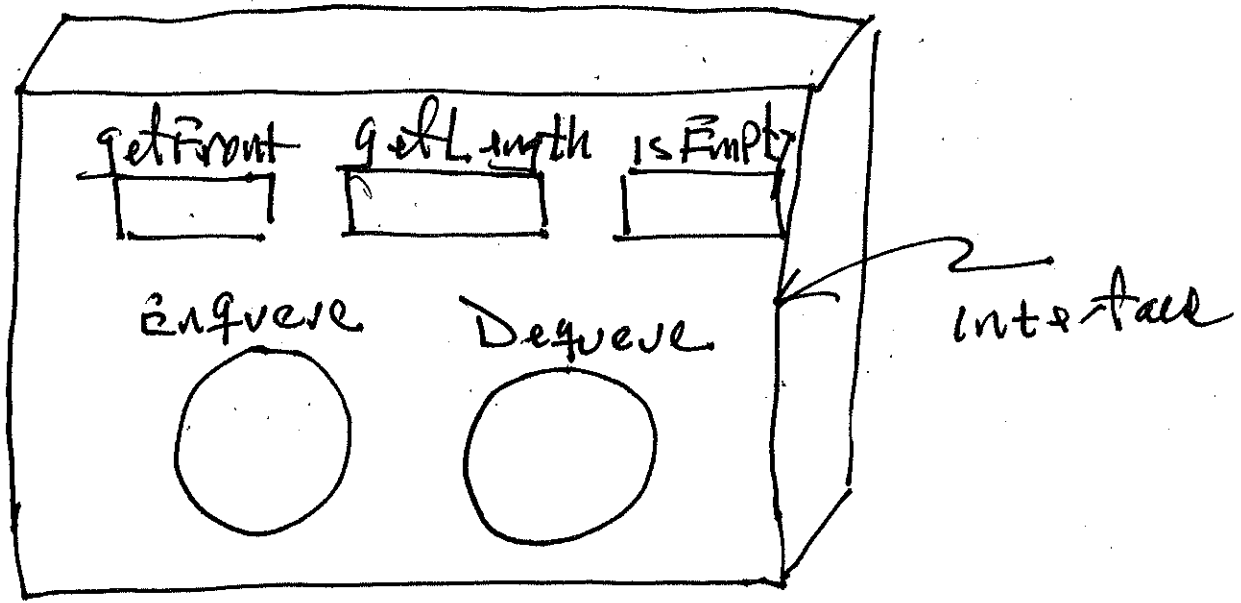
what's the error message?

ADT: Operation: Precond.

Ex.

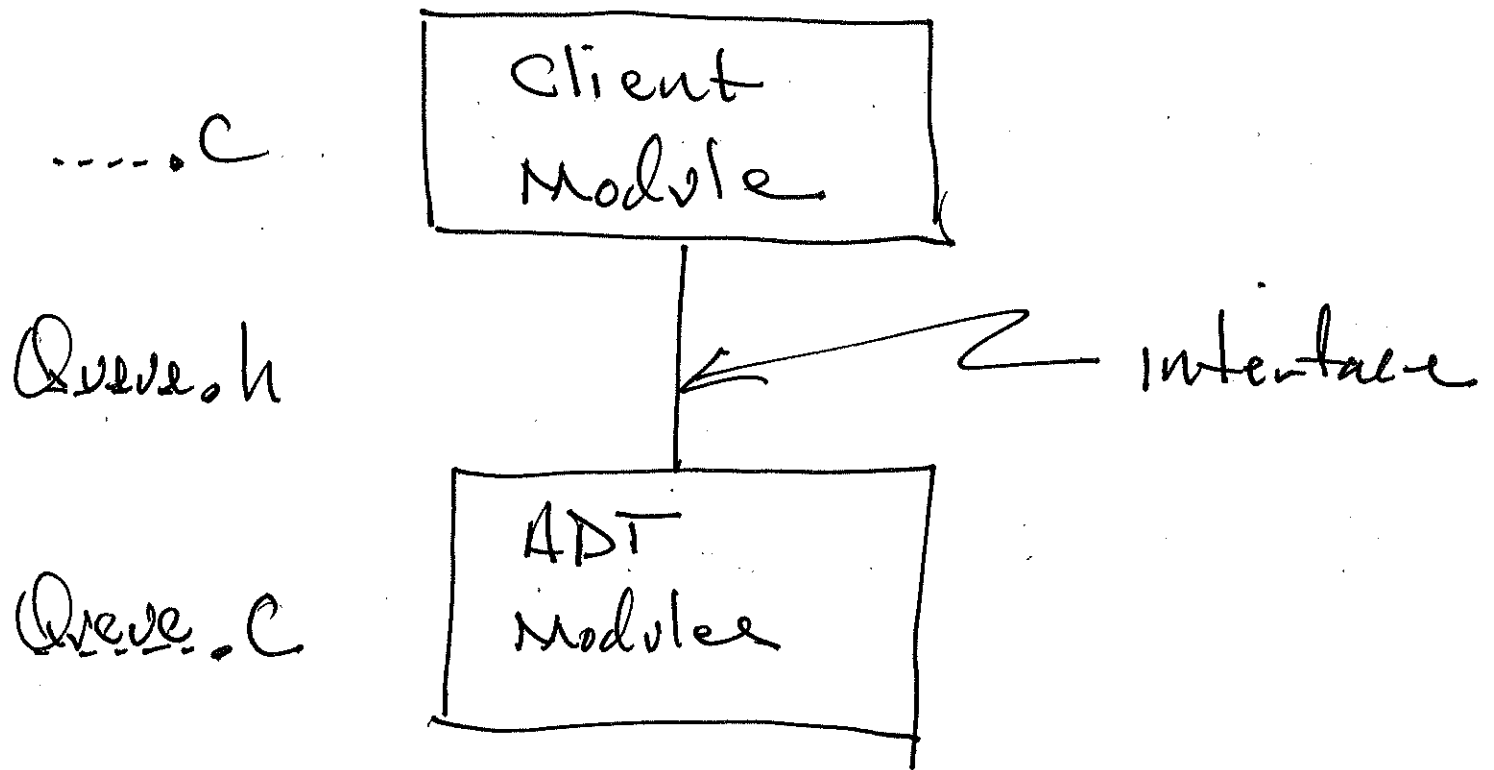
"Queue: getFront(): empty queue"

Black Box Picture:



structure chart

16



Information hiding:

client only sees the front
Panel of the Black Box,
not the inside.

Ex. Queue.c

Queue.h

QueueClient.c