

ADT Handout:

an ADT (abstract data type) consists of 2 things

(1) a set of 'structures' called states: S .

(2) an associated set of operations on states.

- access functions: return info on a state, don't change state
- manipulation procedures: alter the state, don't return.

Ex. Integer Queue

states : finite sequences of integers.

operations :

- manip. Proc. : Enqueue()
Dequeue()
- access fns : getFront()
getLength()
isEmpty()

History of states :

<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
isEmpty()	(1, 7)	false
getFront()	(1, 7)	1

note: if `isEmpty()` is true then what should `getFront()` and `Dequeue()` do?

ans. establish Preconditions that must be true before an op. can be called.

Both `getFront()` and `Dequeue()` have pre. cond. `not isEmpty()`

ADT ops should always check each pre.cond.

```
if (pre cond 1 violated) {
```

```
    print error-msg.
```

```
    exit prog.
```

```
}
```

```
if (pre cond 2 violated) {
```

```
    print error
```

```
    exit
```

```
}
```

```
...
```

```
// nominal execution...
```

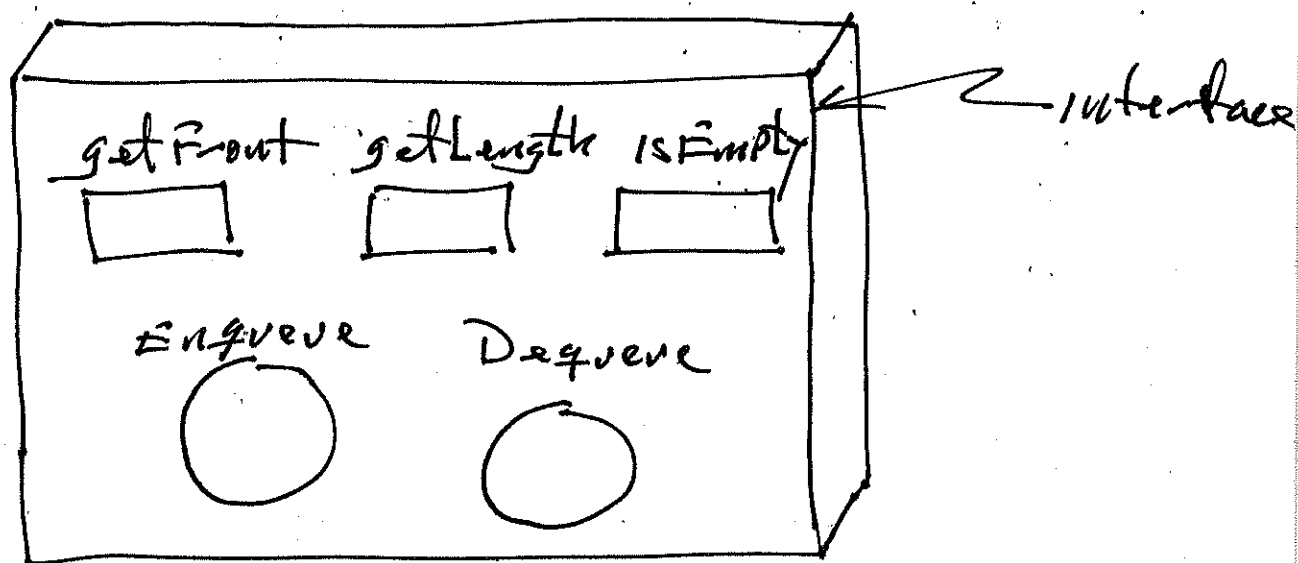
what does error msg. say?

ADT: OP: Pre. Concl. (what's wrong)

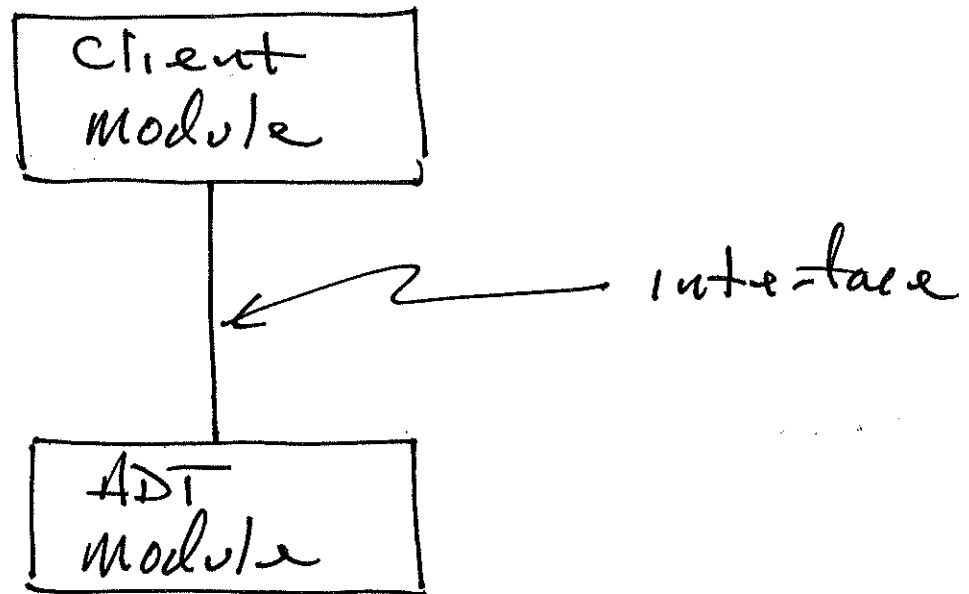
Ex. (Queue ADT)

"Queue: getFront(): empty Queue"

Black Box Picture:



Structure chart :



information hiding : client does not see inside of black box.