cse 101 1-9-24

An ADT (abstract data type)

Consists of 2 things

(1) A set S = {states}

(2) an associated set of operations on states.

- access functions
  return information about $s \in S$.
- manipulation Procedures
  alter state

Ex. Integer Queue

- $S = \{$ finite sequences of ints $\}$
- operations:

access fcns

getFront()
getLength()
IsEmpty()

Manipulation Procedures

Enqueue()
Dequeue()

state: (x x x x x)

front ↑ back ↑

History of states

op	state	return
	( )	–
Enqueue(5)	(5)	–
Enqueue(1)	(5, 1)	–
Enqueue(7)	(5, 1, 7)	–
Dequeue()	(1, 7)	–
Enqueue(3)	(1, 7, 3)	–
getLength()	(1, 7, 3)	3
getFront()	(1, 7, 3)	1
isEmpty()	(1, 7, 3)	false

Preconditions:

dequeue() has precondition
not isEmpty()

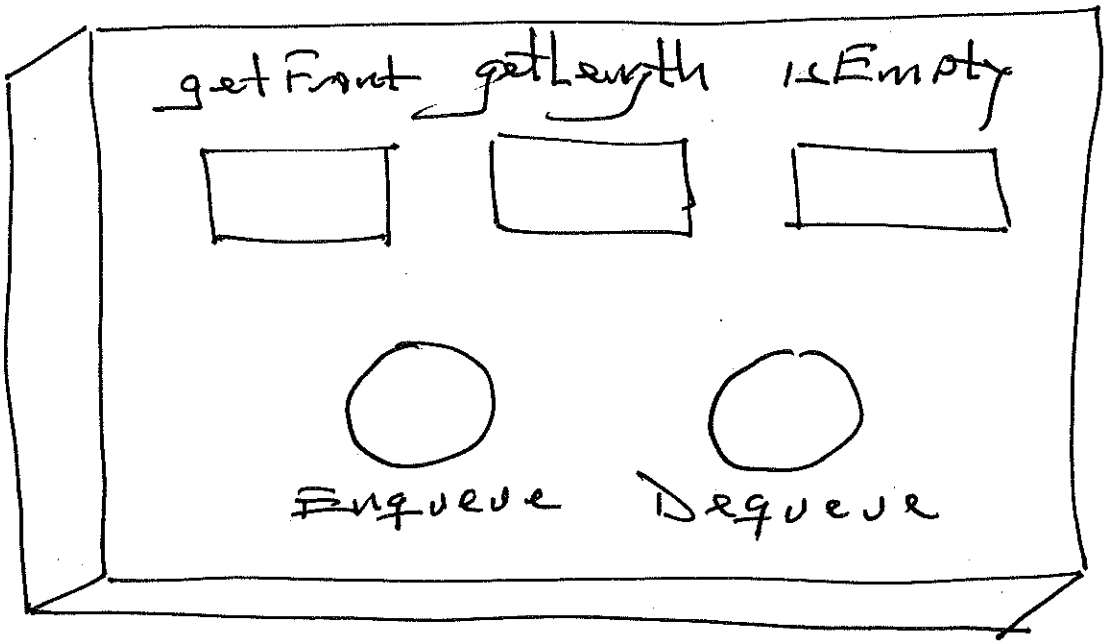
getFront() " " not isEmpty()

Policy (in C):

- If a Pre. cond. is violated, kill program with error msg.

  what ADT: what OP: what Precond.

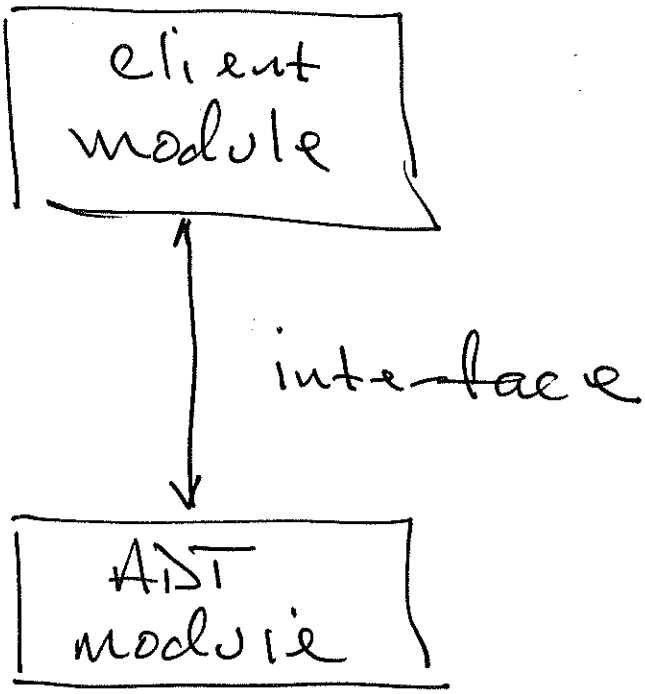
Black Box Picture:



"client" does not see inside of black box.

Information hiding.

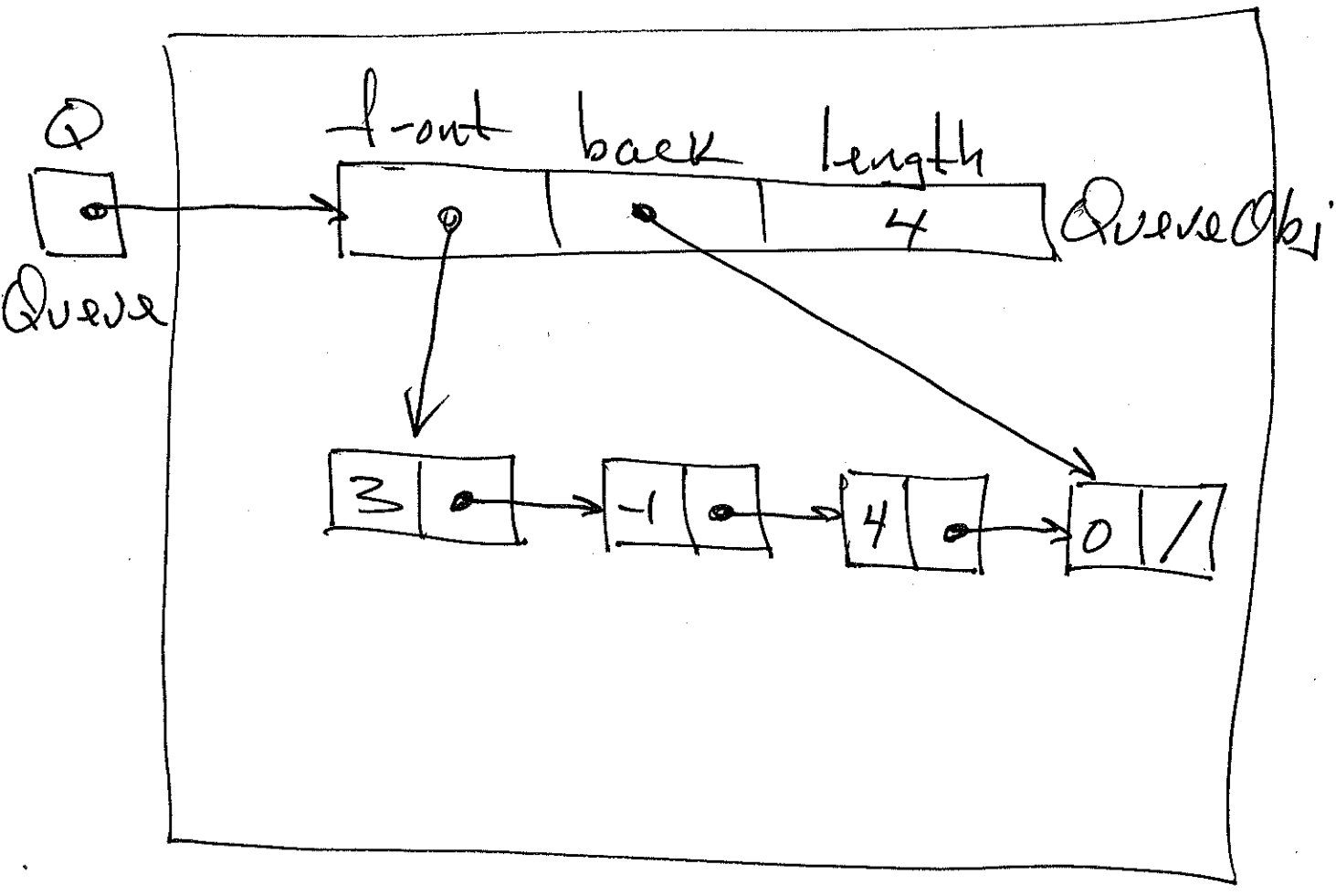
Modules



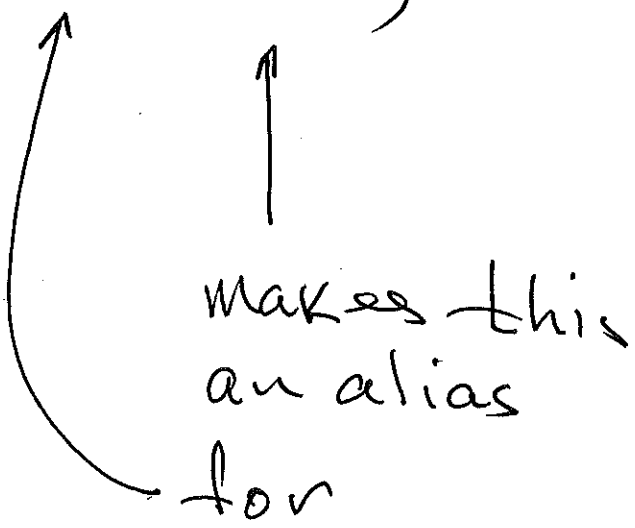
Queue : client picture

(3, -1, 4, 0)

Queue : inside picture (C lang.)



typedef A B;



typedef struct{.....} blah;