CSE 101-02 4-4-23

Handout on ADTs in C.

Abstract Data Types

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

(2) a set of operations on states:

- manipulation procedures:
  change the state of an instance.
- access functions
  return information on a state.

Ex. Queue ADT (of integers)

states: finite sequences of ints.

		len
e.g.	(5, 1, 2, 7, 3, -2, 5)	7
	(11)	1
	( ) empty state	0

operations:

Enqueue()
Dequeue()
getFront()
getLength()
isEmpty()

history of states:

OP.	state	return
	( )	-
Enqueue(5)	(5)	-
Enqueue(1)	(5, 1)	-
Enqueue(8)	(5, 1, 8)	-
Dequeue( )	(1, 8)	-
getFront()	(1, 8)	1
getLength()	(1, 8)	2
isEmpty()	(1, 8)	false
⋮	⋮	⋮

Note:

Dequeue()

getFront()

are not defined on empty state().

Precondition

establish the set of states

to which op. can be applied.

for Both Dequeue(), getFront()

!isEmpty()

Policy:

if a precond. is violated, print an error msg, then Quit Program.

error msg:

- what ADT
- what operation
- what precondition

state precond. in a comment block before operation.

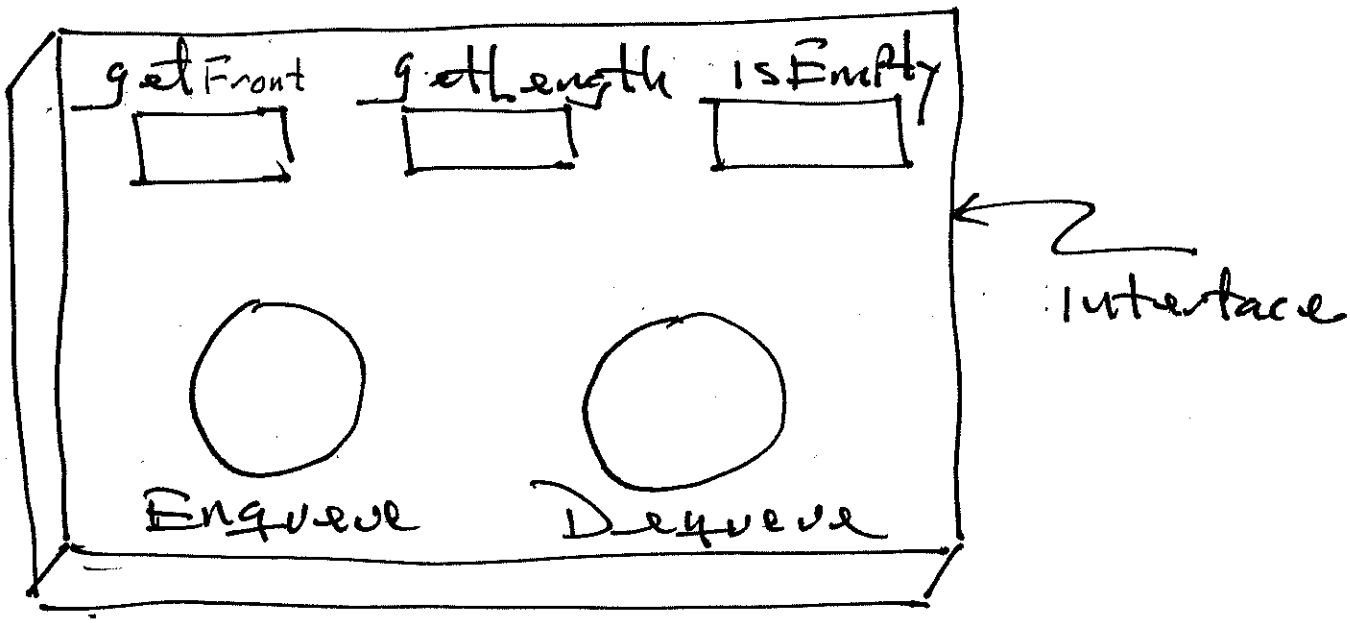
Calling an operation:

in C : Dequeue(Q) ← inst. of Queue ADT.

in C++ : Q.Dequeue()

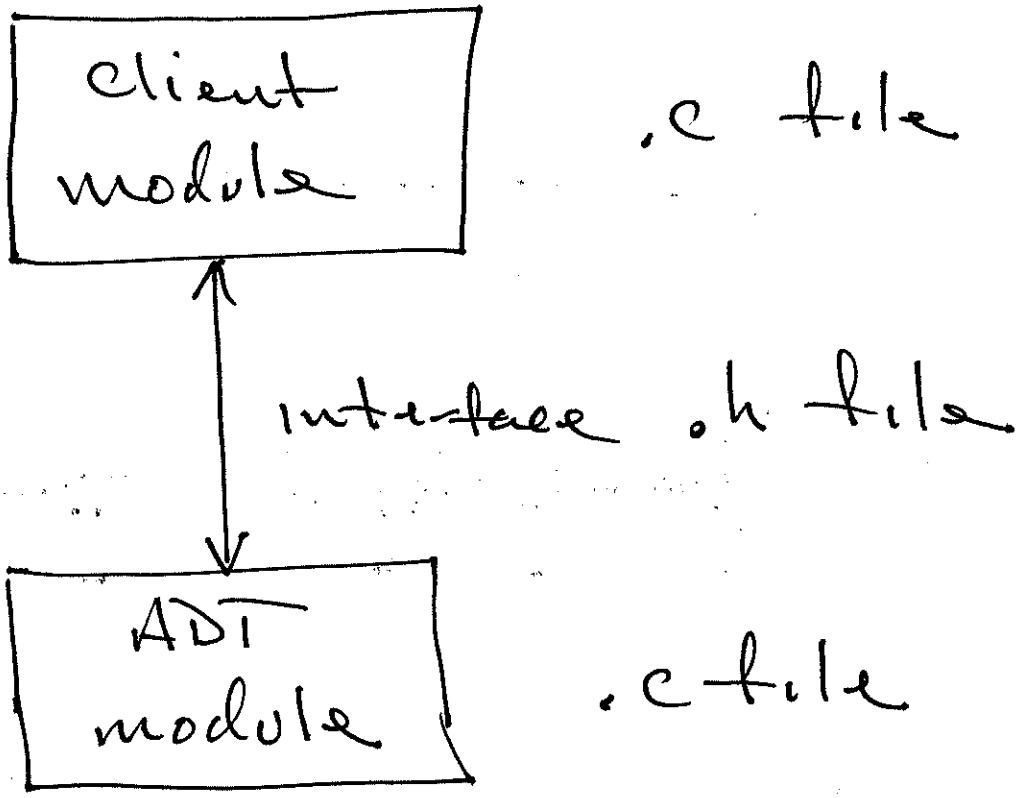
↑ implicit 1st arg.

Black Box Picture:

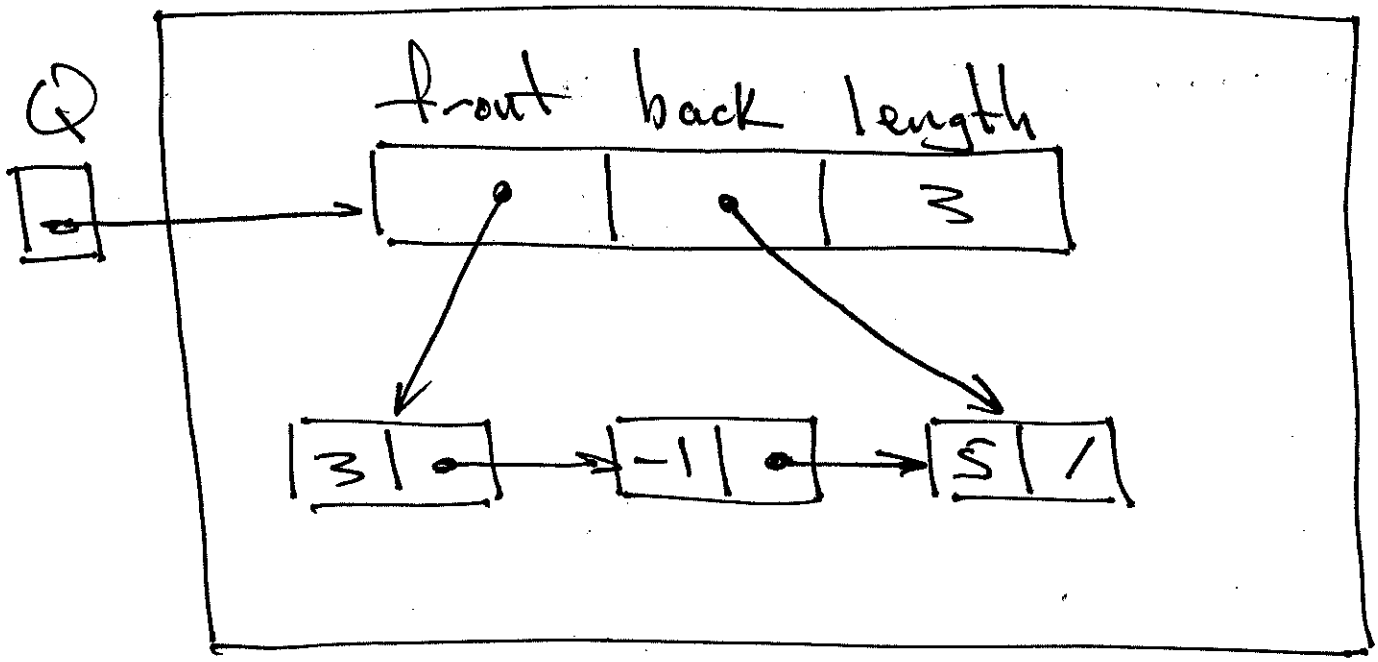


Information hiding:

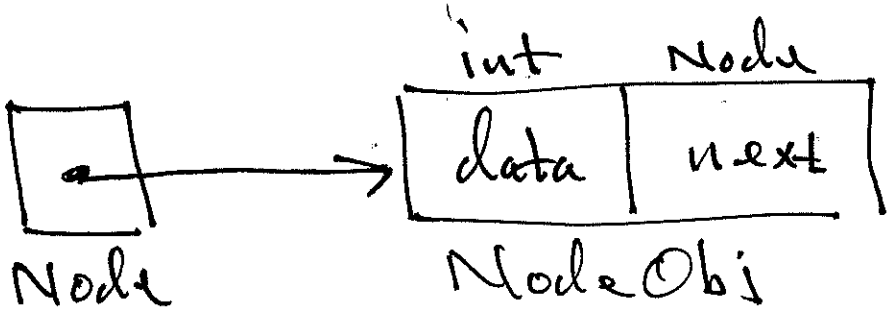
The client interacts with an ADT instance only through the interface



Inside of the Queue black box. (linked list)



client view: (3, -1, 5)



How to create a Node in C?

```
typedef A B;
```

↑ B is now an alias for A.

```
typedef struct NodeObj{
    int data;
    struct NodeObj* next;
} NodeObj;
```

A

B

```
typedef NodeObj* Node;
```

A B