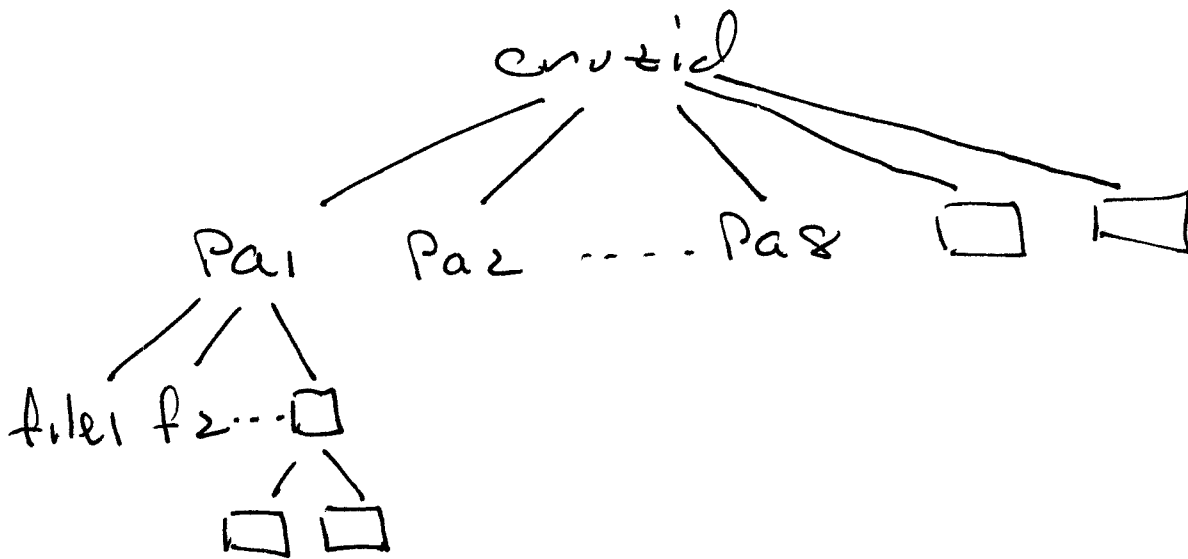


CSE 101 4-2-20

1

Gitlab @ UCSC

CSE101 / Spring-20

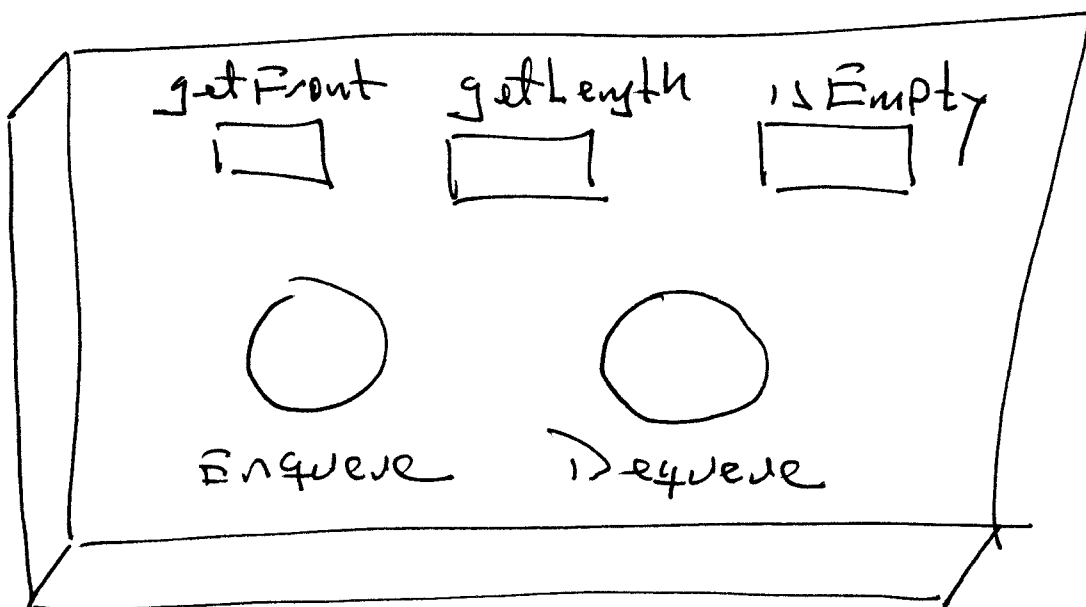


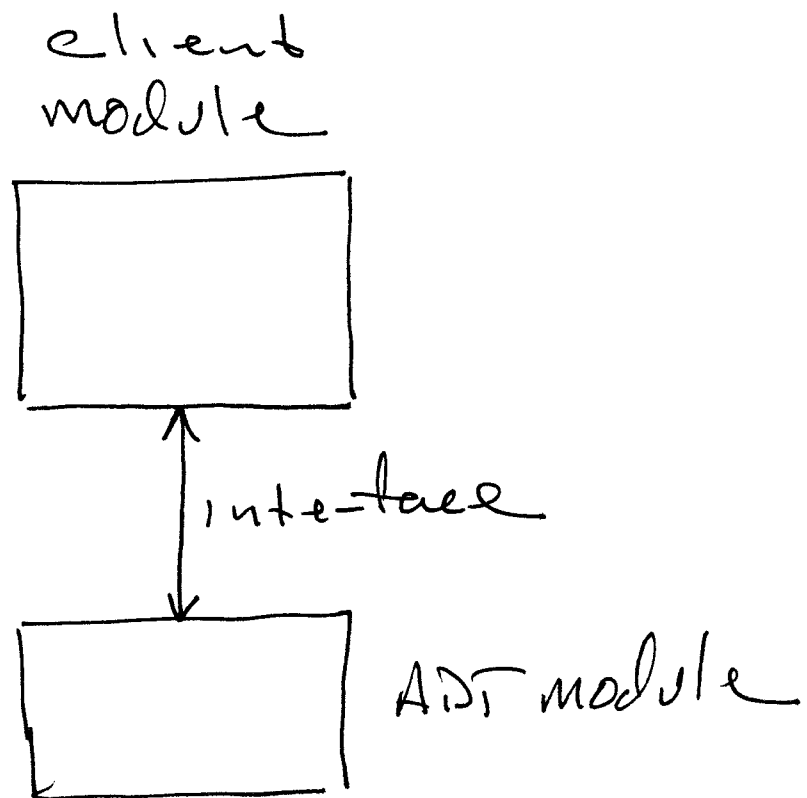
continue ...

<u>op</u>	<u>state</u>
Dequeue()	()
↖ undefined on ↗	

Dequeue() Precondition:
not isEmpty()

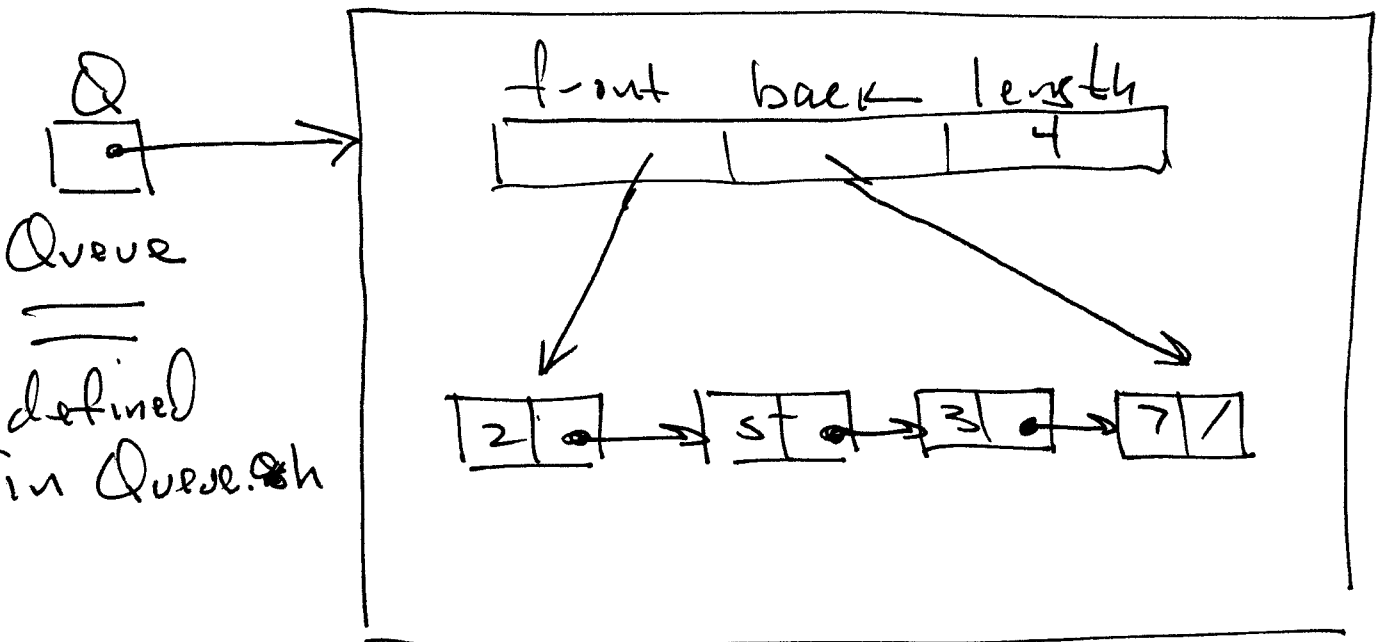
Black Box Picture : Data Hiding





no Private in C
is Private in C++

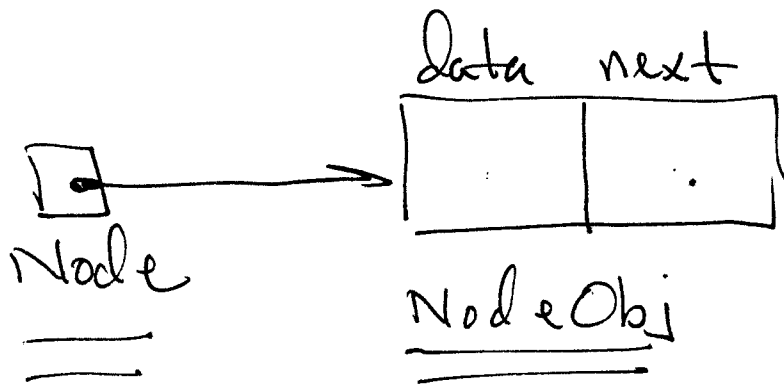
inside view:



Queue object / QueueObj
defined in
Queue.c

client view:

(2 5 3 7)

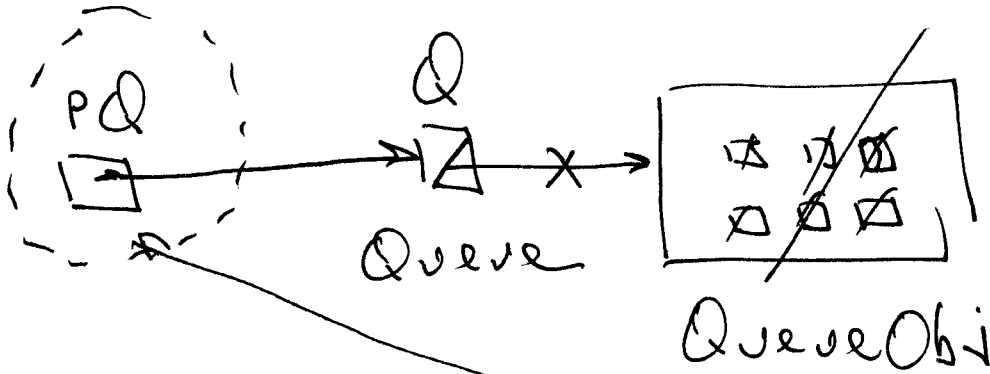


Both defined in Queue.c

syntax for typedef

typedef A B;

B is now an alias
for A



```
void freeQueue(Queue* PQ) {
    // free heap memory assoc. with *PQ
```

```
    free(*PQ);
    *PQ = NULL;
}
```

this is why we need added level of indirection .

Policy on Preconditions:

if a pre cond. is violated when client calls an ADT op then ADT must.

- Print error message
 - what ADT module
 - what ADT op
 - what precondition

- Kill Program.

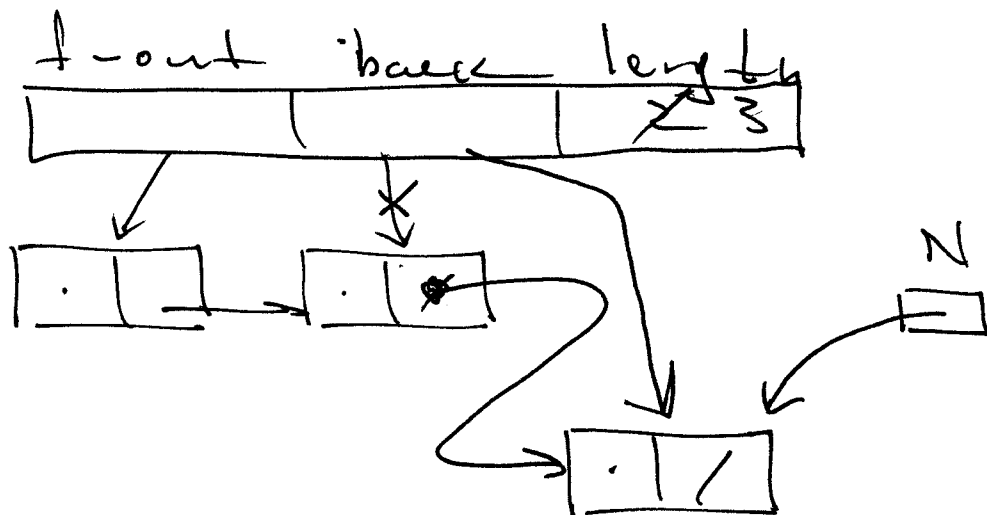
style

```

type ADT_OP(.....){
    if (Pre 1 violated){
        :
        exit(1);
    }
    if (Pre 2 violated){
        :
        exit(1);
    }
    // nominal execution
    :
}

```

Enqueue



to

$Q \rightarrow \text{back} \rightarrow \text{next} = N$

$Q \rightarrow \text{back} = N$

Pa4

client view:

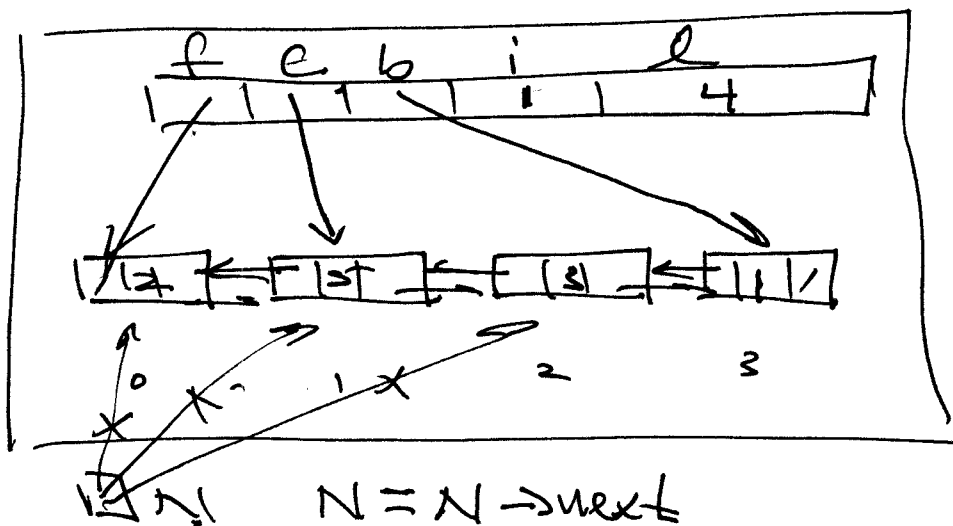
$L = (2 \quad \underline{5} \quad 3 \quad 1)$

delete(4)



$L = (2 \quad 3 \quad 1)$

inside view:



timeshare

