

Thursday Sep. 6
Lecture I

Client vs. Supplier in OOP

```
class Microwave supplier
private boolean on;
private boolean locked;
void power() {on = true;}
void lock() {locked = true;}
void heat(Object stuff) {
  /* Assume: on && locked */
  /* stuff not explosive. */
} }
```

```
class MicrowaveUser {
public static void main(...) {
  Microwave m = new Microwave();
  Object obj = ???;
  m.power(); m.lock();
  m.heat(obj);
} }
```

client

pre-state → client's obligations → supplier's benefits are met.
m. heat(obj);
→ *post-state* (supplier's obligation) → client's benefits are met.

Context object

call in the context of class MU

Requirements of Bank

REQ1: Each account is associated with the *name* of its owner (e.g., "Jim") and an integer *balance* that is always positive.

REQ2: We may *withdraw* an integer amount from an account.

P
↓
instructions followed

Q
↓
service provided

P	Q	
T	T	Green bar
T	F	Red bar
F	T	Green bar
F	F	Green bar

is the contract honored?
 $P \Rightarrow Q$

← contract violation

even though T logically,
some kind of "error"
should be reported
about the client's
faults.

class PointTester } client.

main(- - -) {

Point P = new Point();

↓
service of
constructing
a new object

}
}