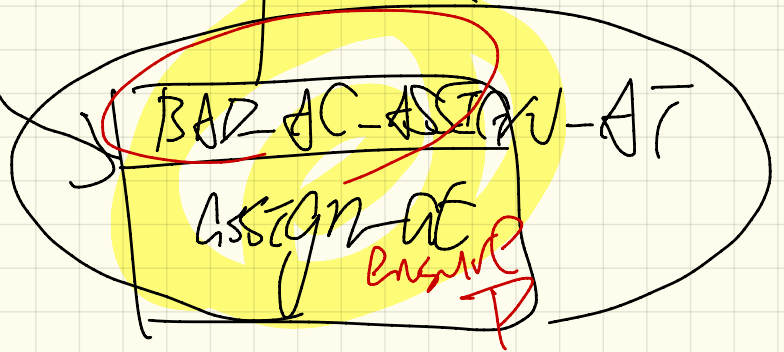
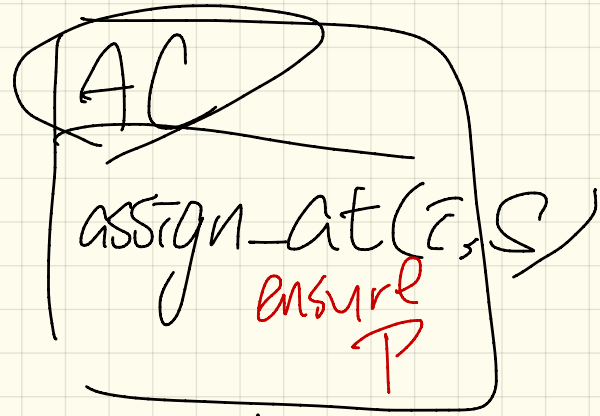


Lecture 5

Monday Sept. 25

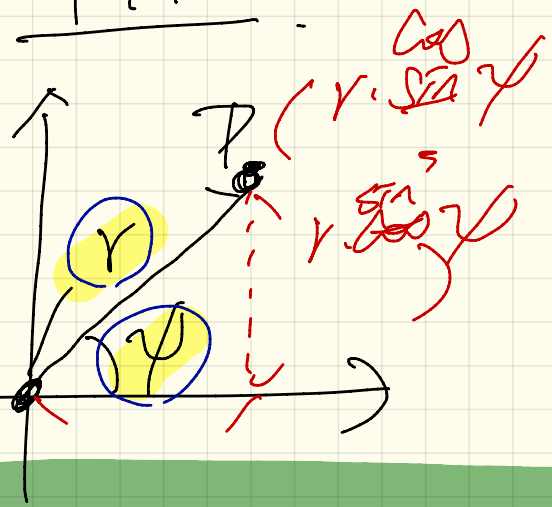
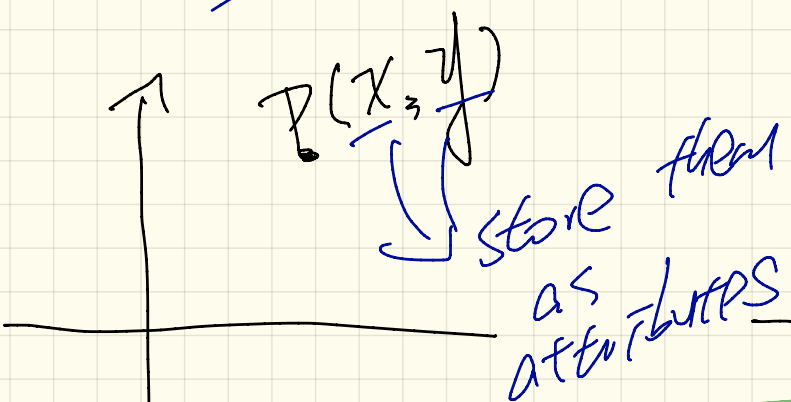


2-D points (Supplier)

$x: r * \cos \psi$
 $y: r * \sin \psi$

Cartesian

Polar



Client: $P(x)$ $P(y)$

```

class Point {
private double x;
private double y;
private double phi;
}

```

change from C++
to Obj.

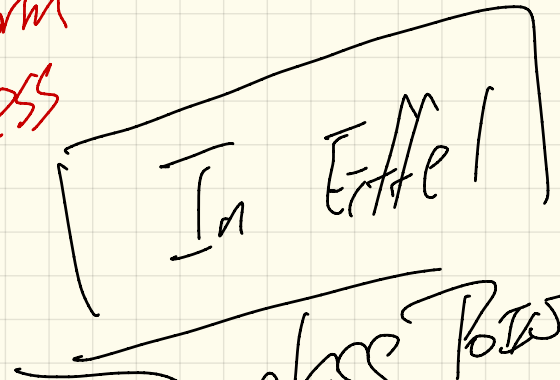
uniform
access

```

double get X() {
    return x;
}

```

x * sin phi

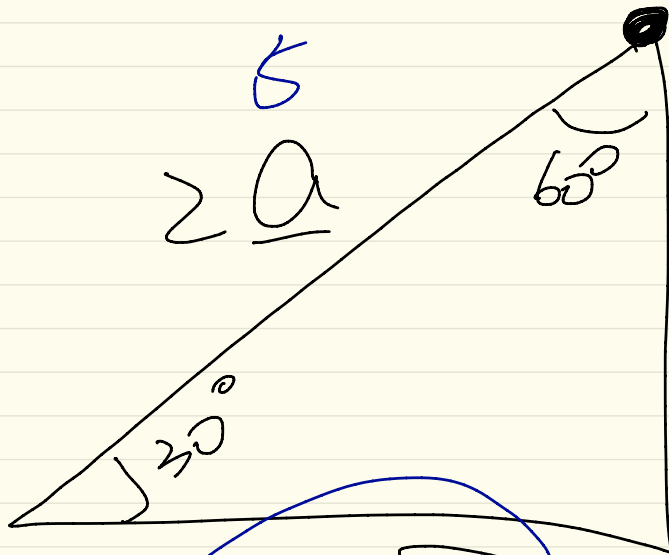


```

class POINT
x: REAL
y: REAL

```

end



$$\frac{a}{\sqrt{3}}$$

$$\left(\frac{2a \cdot \sin 30^\circ}{2a \cdot \cos 30^\circ} \right)$$

a \downarrow
polar system

Cartesian system.
these values
pass right away =
 $(5\sqrt{3}, 5)$

BANK

accounts: ARRAY [ACCOUNT]

balance: REAL

↑ frequent access
↓ less frequent change.

attribute

↓ avoid computation
e.g. large # of accounts.

↓ query.

→ don't have to keep the value in sync for every feature. Every access is a fresh comp.