

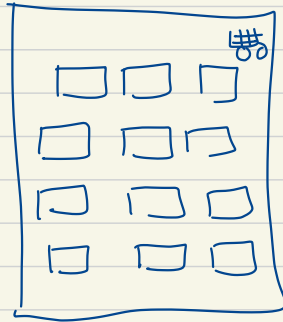
Real world entity Object Oriented Programming

↳ way of looking at real world entities

function ↗ fact
 → fib
 ↳ ✓

Software

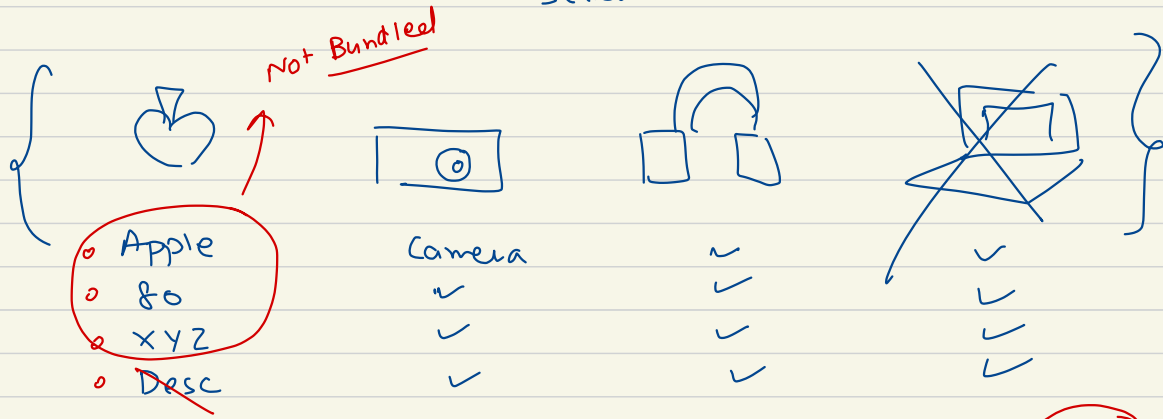
Amazon



- Product
- Deals / offers
- My orders
- wishlist
- Recommendations
- Cart

Product

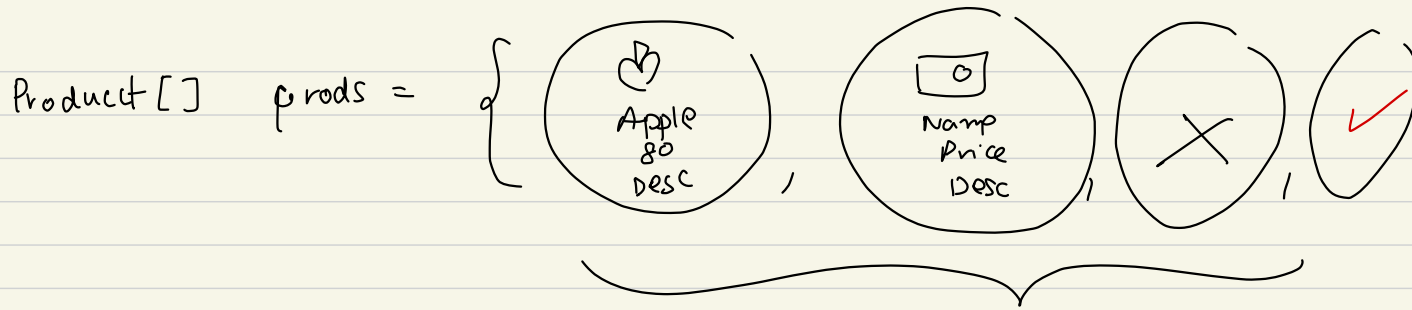
Camera → Name
→ Model No
→ Price
→ Seller



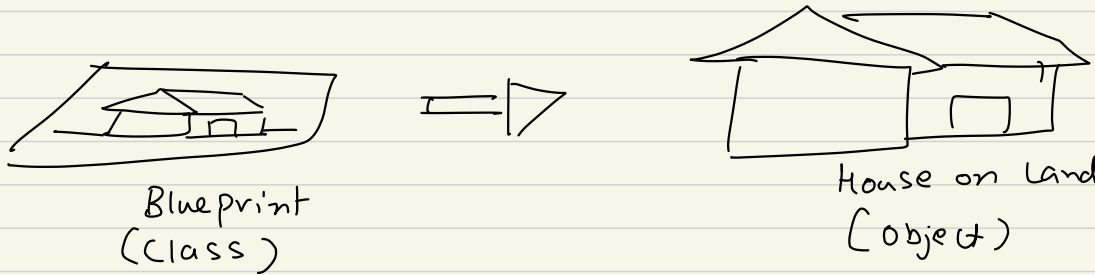
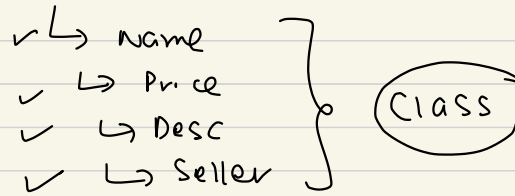
Strings [] names = { "Apple", "Camera", ..., ~~X~~ }
int [] prices = { 80, 10,800, ..., ~~X~~ }
String [] desc = { —, —, —, —, —, —, ~~X~~ }

idx

Difficult



Each product follows template / Blueprint



piece of code

Memory
X

```
class   Product {  
    ✓ String name; ✓  
    ✓ int price; ✓  
    ✓ String desc; ✓  
    ✓ String seller; ✓  
};
```

Objects

Knows
something
about
itself

(Instance Variables/
Data members)

Can also
some
behaviour
(Methods
inside
class)

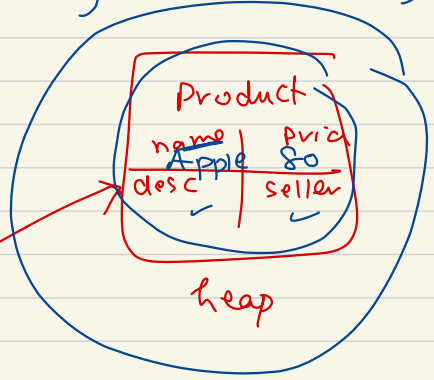
{ p1.price = 80,

Product p1 = new Product();
p1.name = "Apple"

Product p2 = new Product();

Objects take
up memory

p1
└─┬─┘
Slack



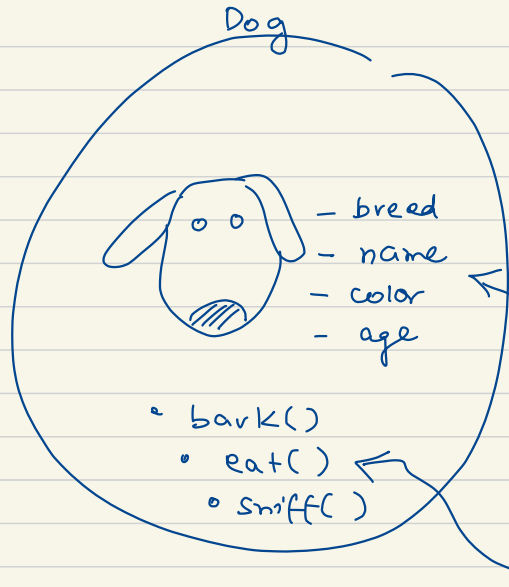
Java

class SomeName {

```
    public static main {  
        ||  
        ||  
        ||  
    }  
}
```

~~SomeName obj = new SomeName();~~ ✗

Scanner sc = new Scanner(); ✓



Example

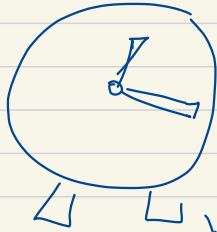
→ Should Knows (Data)

→ Can Do (Behaviour)

- bark()
- eat()
- Sniff()
- run()
- Swim()
- jump()

Real World

• Alarm



→ Data

→ current_time → Time format

→ start_time : ✓

→ sound

→ mode

Behaviour

↳ set Alarm ()

↳ Snooze ()

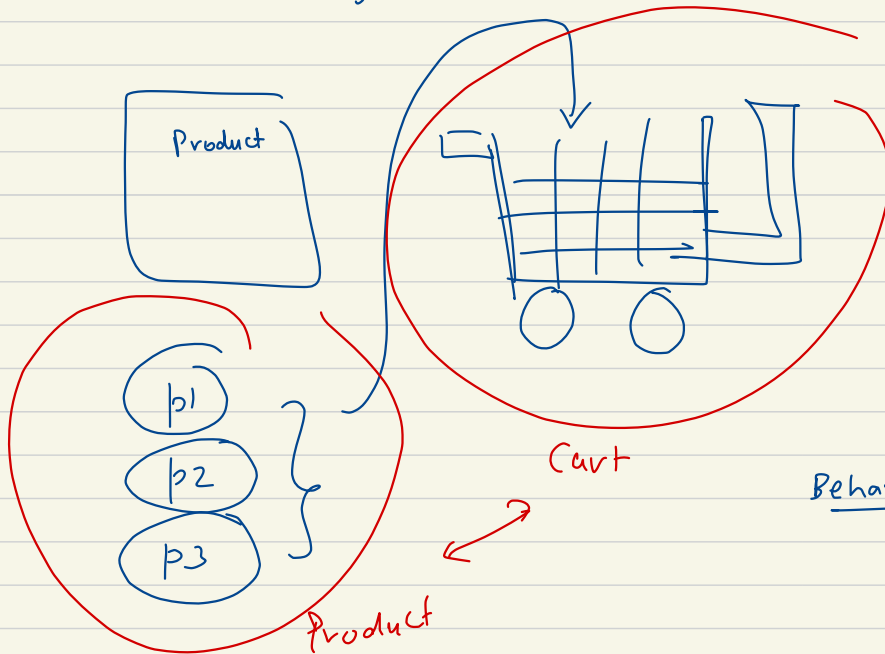
↳ stop Alarm ()

↳ remove Alarm ()

Amazon

Classes , object

Objects do interact with each others



$$1 \times 100 + 2 \times 100 + 1 \times 300 = \text{total}$$

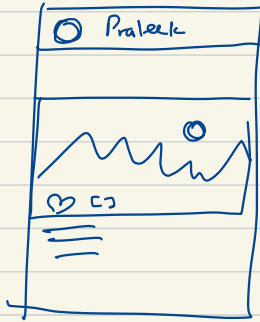
Cart { ① ② ①
 ↓ ↓ ↓
 100 200 300

List<Product> ordered ;
List<int> qty;
int total; ←
int item-cnt;
int tax-%;

Behaviour

add Product() ;
remove Product();
Apply Coupon();
Upd.qty()

Instagram



User

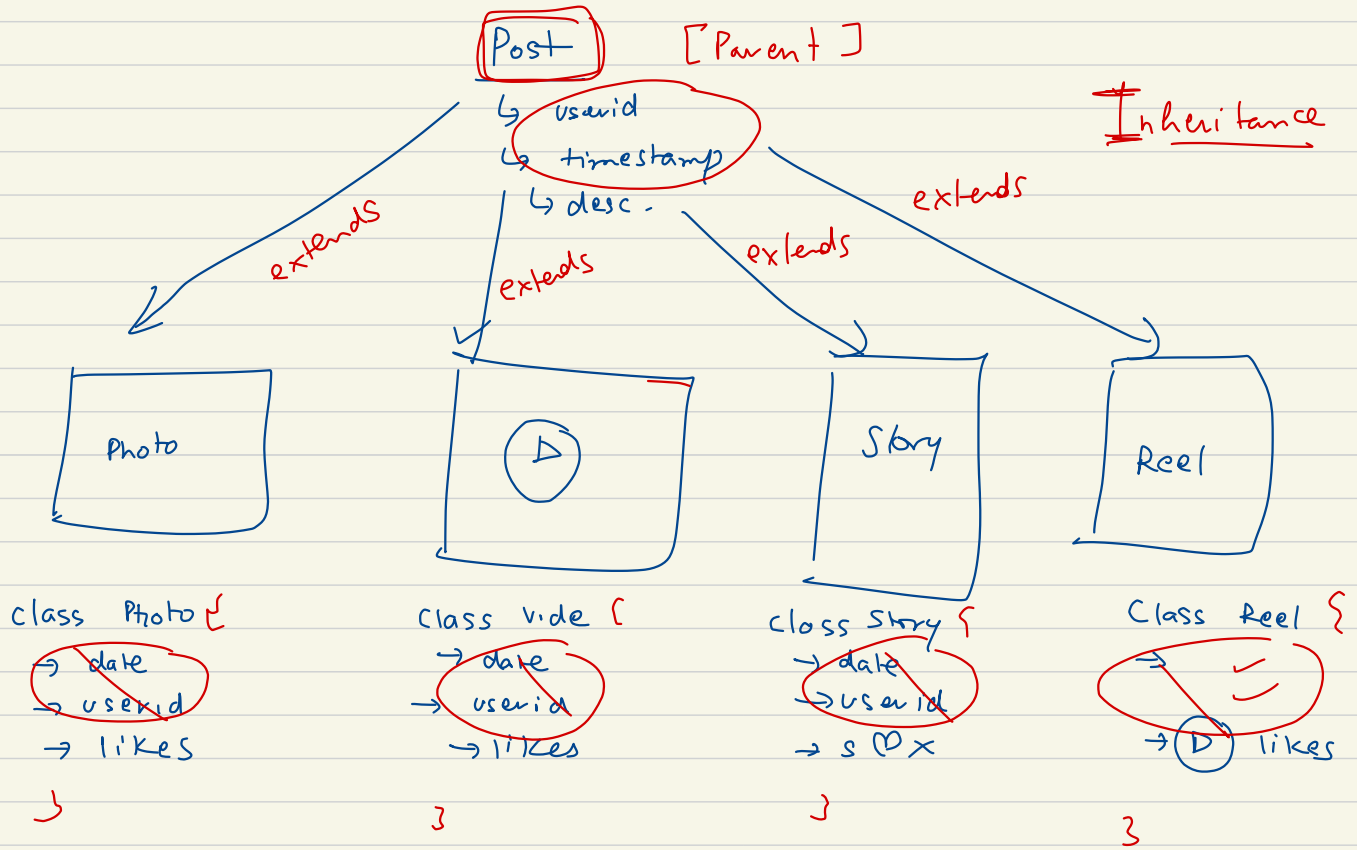
- ↳ Profile Pic
- ↳ id
- ↳ handle
- ↳ bio
- ↳ #followers list <users> followers,
- ↳ list <users> following ✓
• add (---),

Behaviour

- ↳ create Post ()
- ↳ like Post ()
- ↳ Share Story ()
- ↳ Chat ()
- ↳ Audio Call ()
- ↳ Video Call ()
- ↳ Follow ()

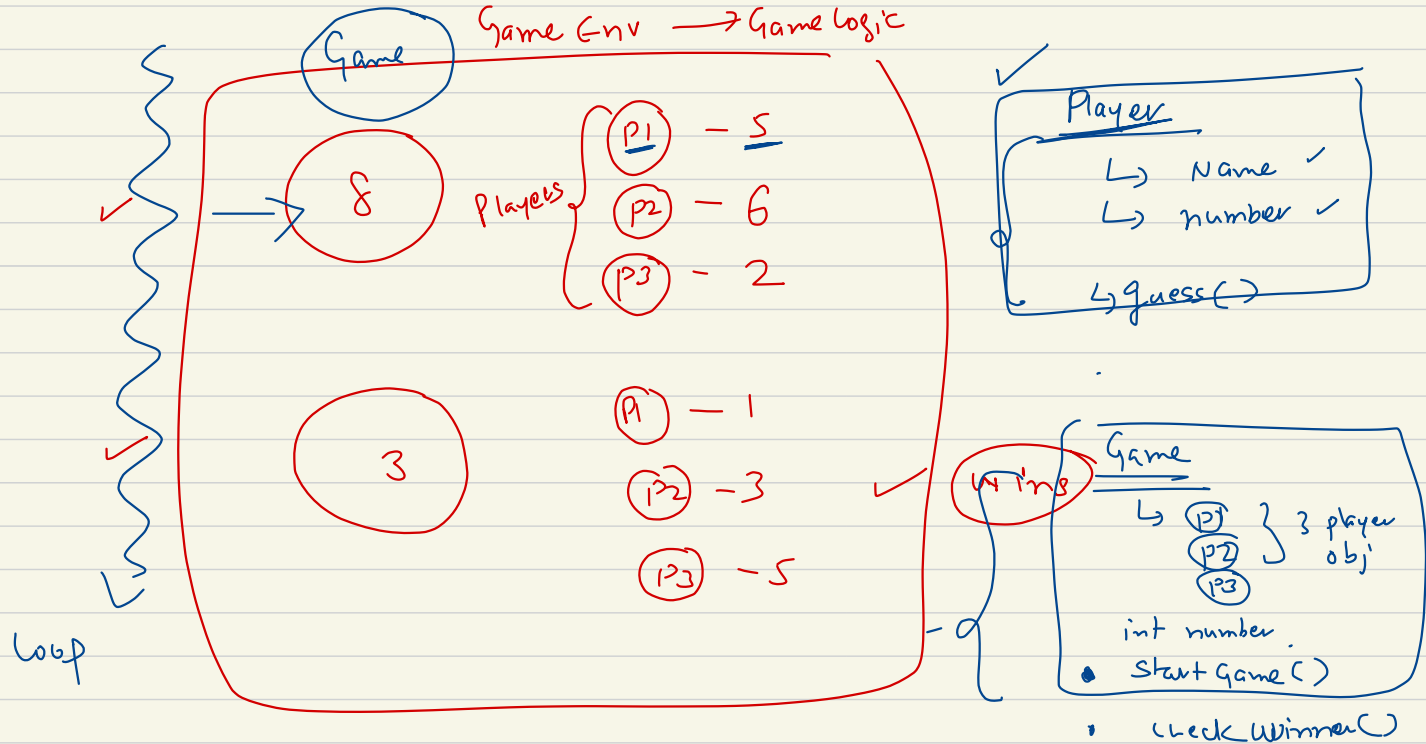


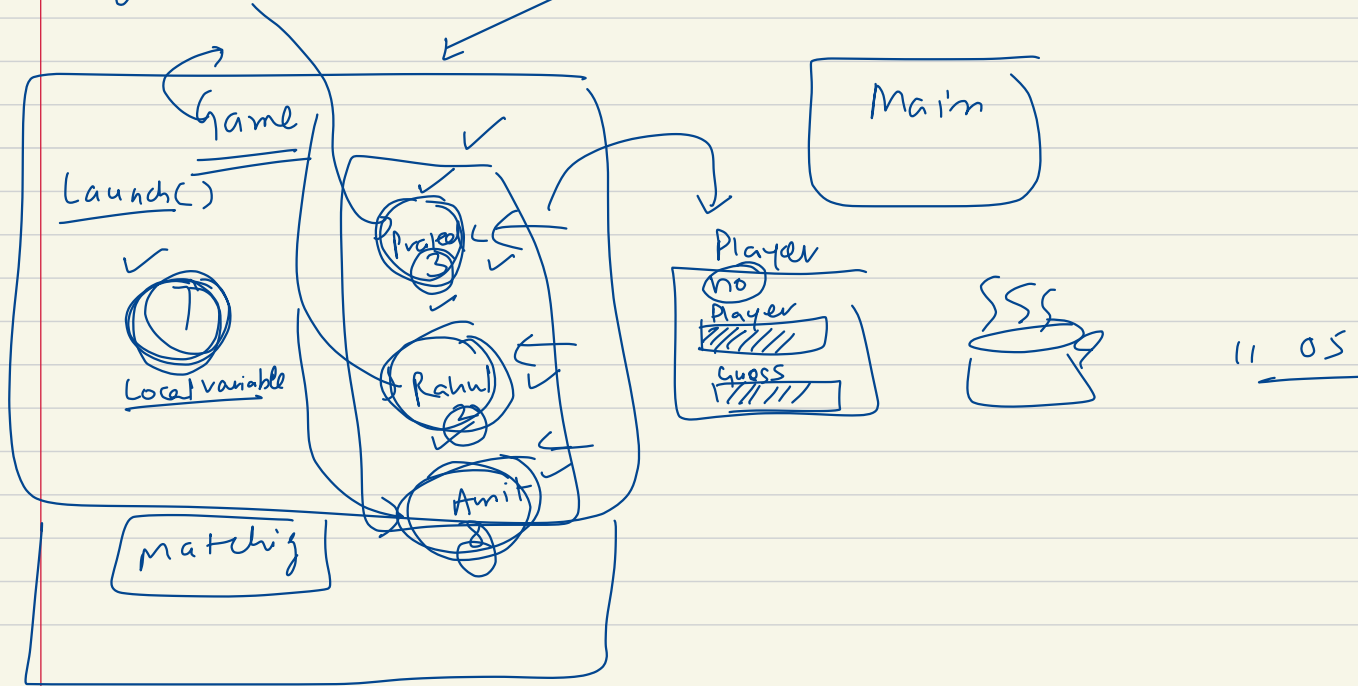
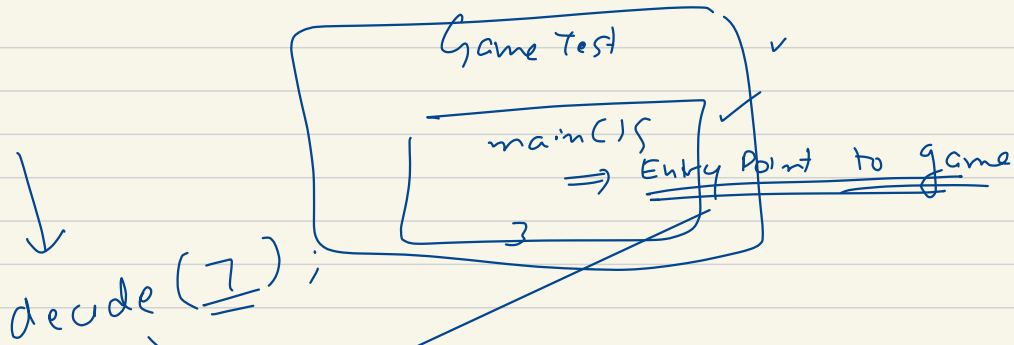
child class



Game

Goal : Create a guess game, in which the system will generate a number from 0 to 9. There are 3 players in the game, each person will make a guess about the number. The first one to guess it correctly wins the game, else everyone tries again.





• { 10, 8, 2, 6, 5 }

• { 2, 5, 6, 8, 10 } $\xrightarrow{\hspace{2cm}}$

Algorithms
x
[Adv Module]

Sorting

Inbuilt Methods

↳ Arrays

↳ ArrayList

