



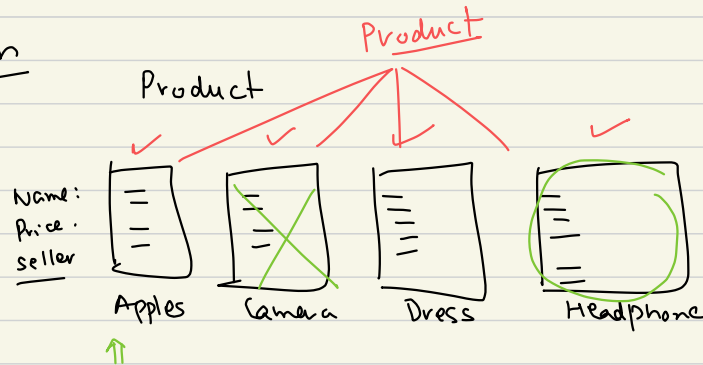
# OOPS

→ Functions / methods

→ New Style of Writing Code

↳ Represent real world entities objects

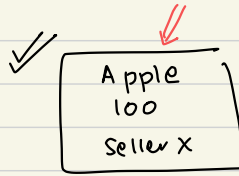
Amazon



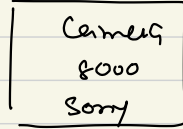
✓ String[] product-names = ["Apples", "Camera", ...]

✓ int[] product-prices = [400, 800, ...]

✓ String[] product-seller = [someSeller, ...]



Product  
Apple  
[object]



Camera  
[object]

```
class Product {  
    String name;  
    int id;  
    int price;  
    String seller;  
};
```

class

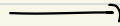
→ takes space  
only in  
code

Class

→ Blue print of what properties & behaviour  
would be present inside  
a class.



Blueprint  
(plan)



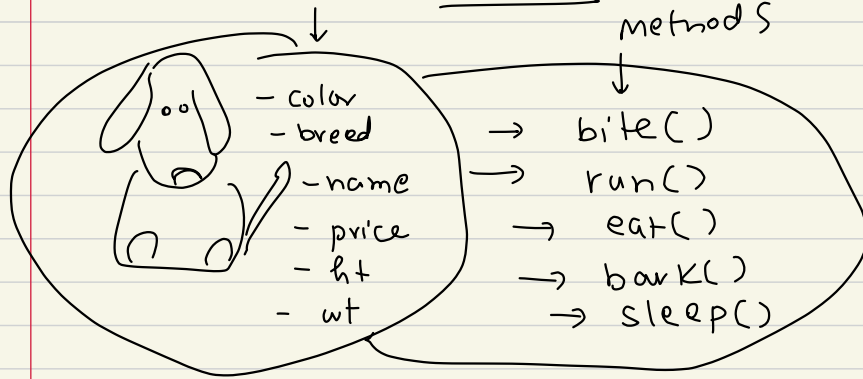
Object

# Object / Instance

↳ Data / Instance variables  
(things that object knows about itself)

↳ Behaviour, things that object can do

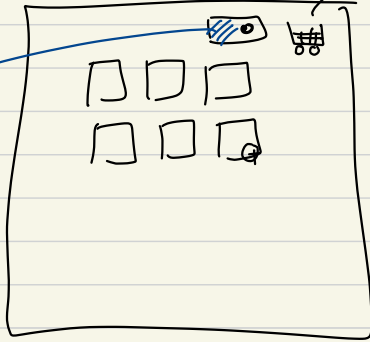
## • Methods



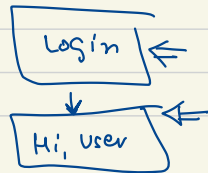
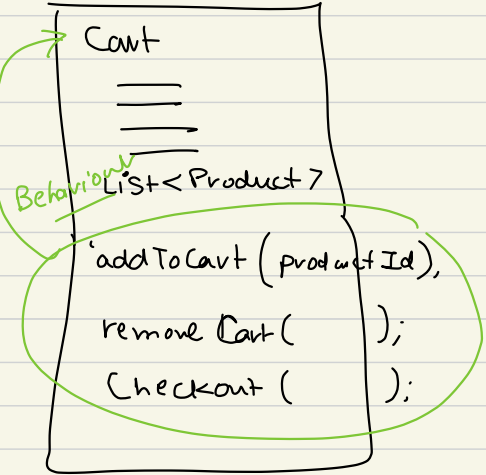
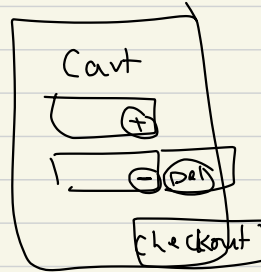
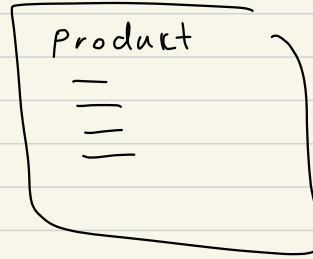
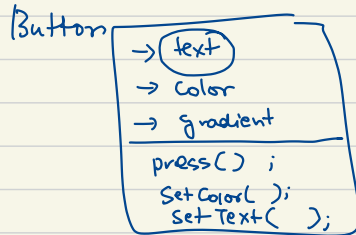
# Amazon / Shopping Site

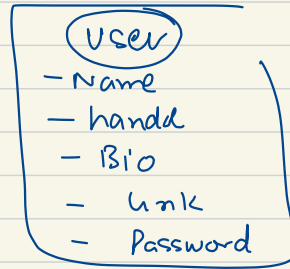
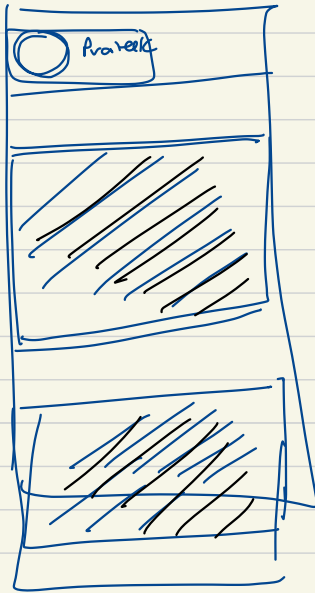
## Software Implementation

UI  
Classes /  
HTML  
objects



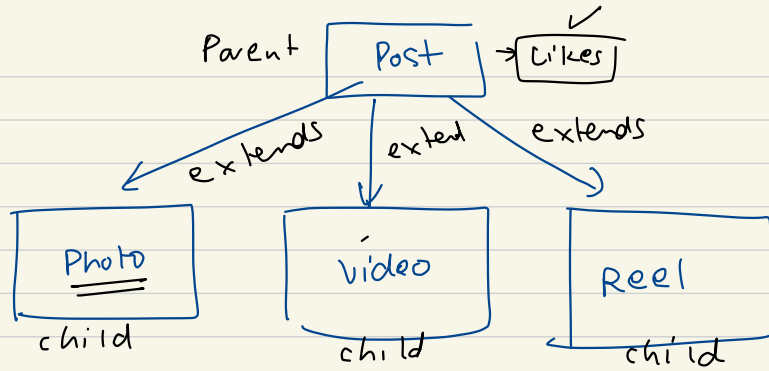
Product {  
  
}





⇒ Post

- date
- id-post
- comments
- likes
- tags
- Description



Inheritance

```
class Photo {  
    → likes  
    →  
    →  
    → x No Audio  
}
```

```
class Video {  
    → likes  
    →  
    →  
    → Music  
}
```

```
class Reel {  
    → int likes  
    →  
    →  
    → Song ⇒  
}
```



class Alarm {

→ Time

→ Type/Mode

→ Start Time

→ Duration

→ Ring Tone

methods

→ set Alarm()

→ remove Alarm()

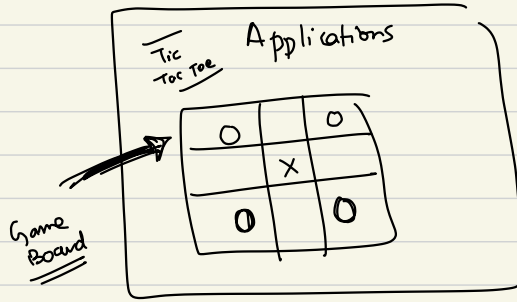
→ Snooze()

→ Dismiss()

}

~~\*~~





2 Players

```
class Player {
    char mark;
    String name;
    Player (String name, char mark) { }
    make A Move (x, y) { = }
}
```

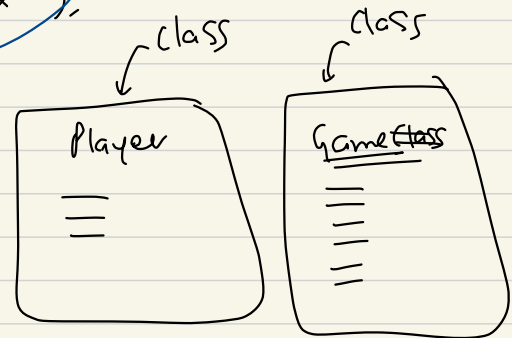
O X

```
Player p1 = new Player ("Prateek", "O")
Player p2 = new Player ("Rahul", "X")
```

```
class GameTic TacToe {
    int board [3] [3];

    StartGame () {
        = : // Init Players,
    }

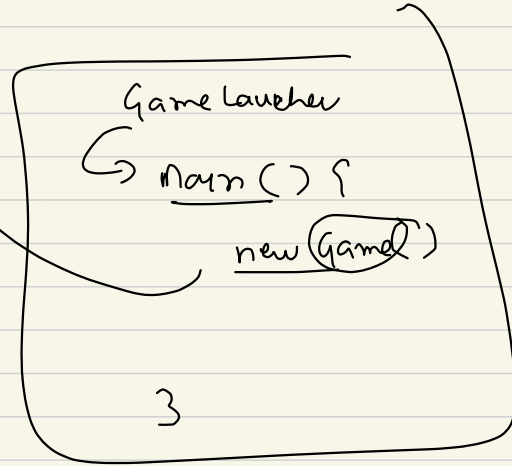
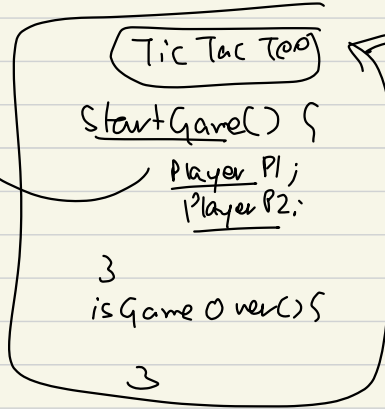
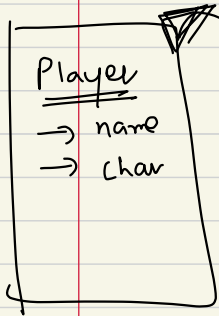
    while (game is not over) {
```



```
GameLauncher () {
    main () {
        g = new GameTicTacToe ();
        g.Start ();
    }
}
```

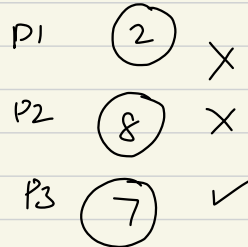
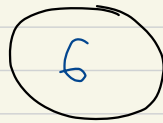
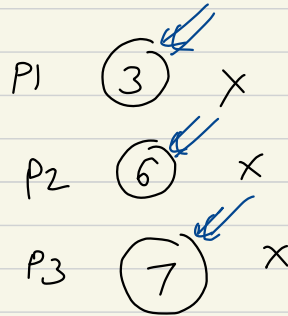
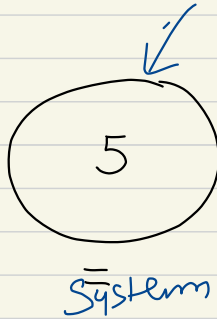
G O O  
X X X

}  
bool isGameOver() {  
    ⇒ checks  
}

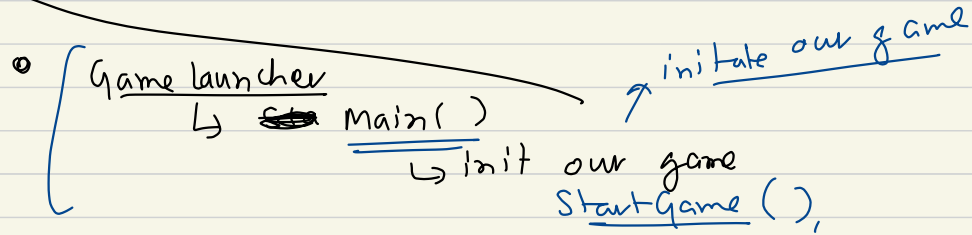
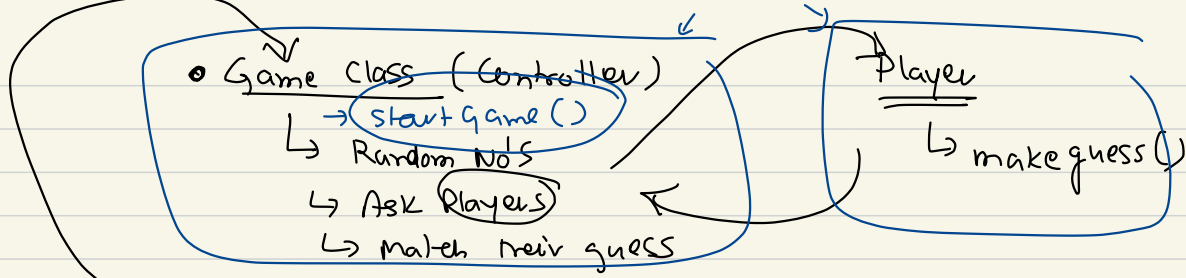


Guessing Game: The game involves 'game' object and 'three' players. The game generates a random number between 0 to 9 and three players try to guess it.

Player who guesses correctly he wins, if all 3 fails the game continues with try again.



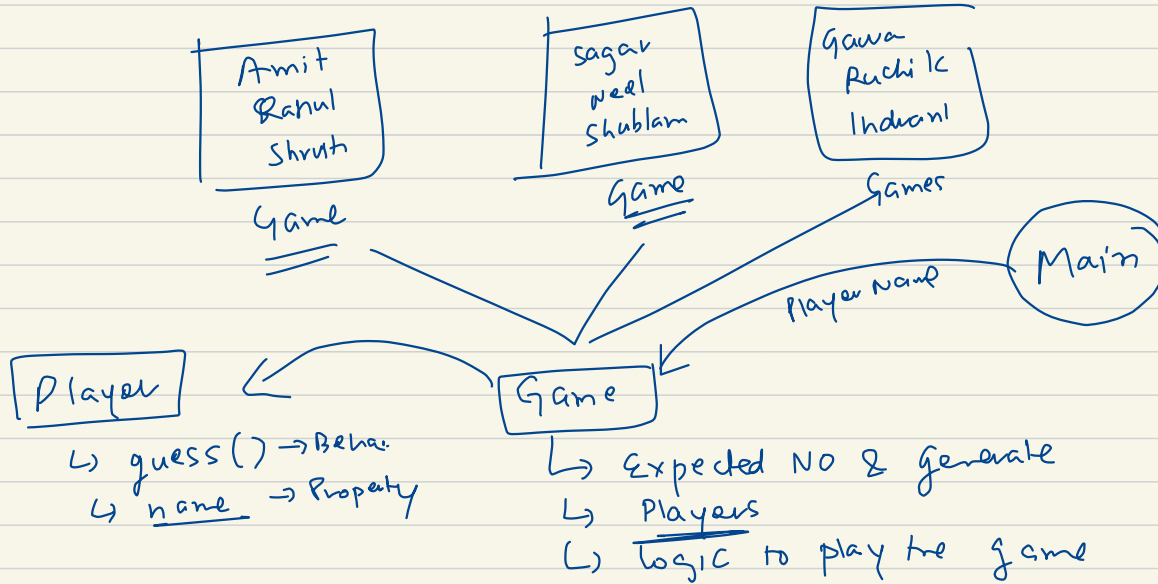
win  $\Rightarrow$  Stop



```

public class GameLauncher {
    public static void main(String[] args) {
        Game guessGame = new Game("Amit", "Rahul", "Shruti");
        guessGame.launch();
    }
}

```



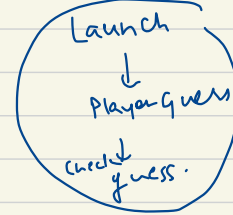
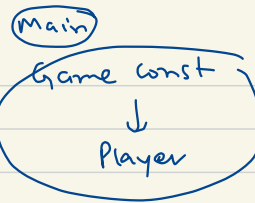
```
class Game{
    Player p1;
    Player p2;
    Player p3;
    int expectedGuess;←
```

①  
init  
players →

```
Game(String name1, String name2, String name3){
    p1 = new Player(name1);
    p2 = new Player(name2);
    p3 = new Player(name3);
}
```

helper

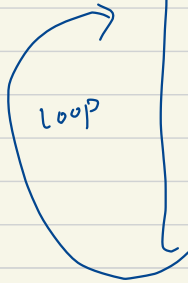
```
x {
    boolean checkWinner(){
        if(p1.number==expectedGuess){
            System.out.println(p1.name + "wins");
            return true;
        }
        else if(p2.number==expectedGuess){
            System.out.println(p2.name + "wins");
            return true;
        }
        else if(p3.number==expectedGuess){
            System.out.println(p3.name + "wins");
            return true;
        }
        return false;
    }
}
```



②



```
void launch(){
    //do we have a winner
    ⇒ expectedGuess = (int)(Math.random() * 10);
    System.out.println("Guess expected is " + expectedGuess);
    while(true){
        → p1.guess(); ✓ 8
        → p2.guess(); ✓ 3
        → p3.guess(); ✓ 2
        //if we we get winner we should stop
        [ if(checkWinner()==true){
            break;
        }
        else{
            System.out.println("Try again");
        }
    }
}
```



Brain  
p  
= 8

```
class Player{
    String name;
    int number; ←
    static Scanner sc = new Scanner(System.in);

    Player(String name){
        this.name = name;
    }

    → { void guess(){
        // take input from the user to make a guess
        number = (int)(Math.random() * 10) ; //sc.nextInt();
        System.out.println(name + " guessed a number " + number);
    }
}
```