

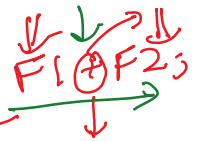
$$F1 = F1 * F2;$$

void

$$F3 = F1 + F2$$

$$\frac{3}{4} + \frac{5}{3}$$

$$F3 = F1 * F2;$$

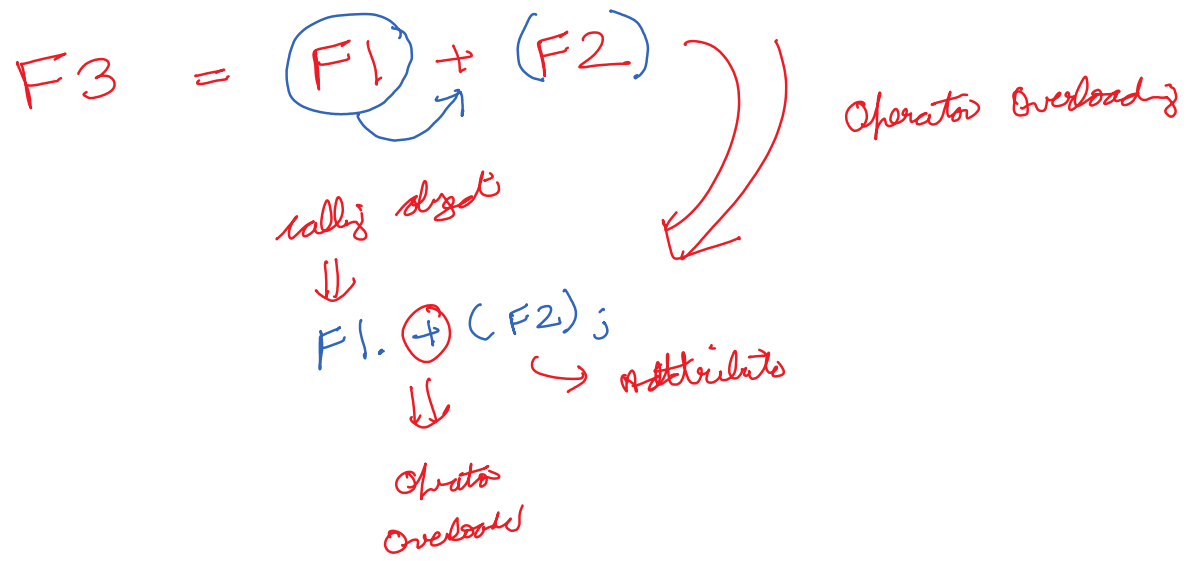


Strig →

Nbr →

Operator Overloading

$$F3 = F1.add(F2)$$



$w = 5;$
 $\text{Pl } \textcircled{1} \text{f3} \rightarrow \text{Prog}$
 $\textcircled{++\bar{w}}$

$\bar{w} = 6j$

$j = ++\bar{w};$
 $\hookrightarrow \textcircled{6}$



chong

$$\frac{5}{3} + 1 = \frac{5+3}{3}$$

splitting

Fracton $F2_1 = \textcircled{++f1j}$

Fracton $F2(F1);$

Fracton $F2 = \textcircled{F1}j$

copy constructor

$\bar{w} = s_j$

$\bar{j} = ++w$
 \downarrow
 \downarrow

into $\bar{j} = ++(++w)$ Execution

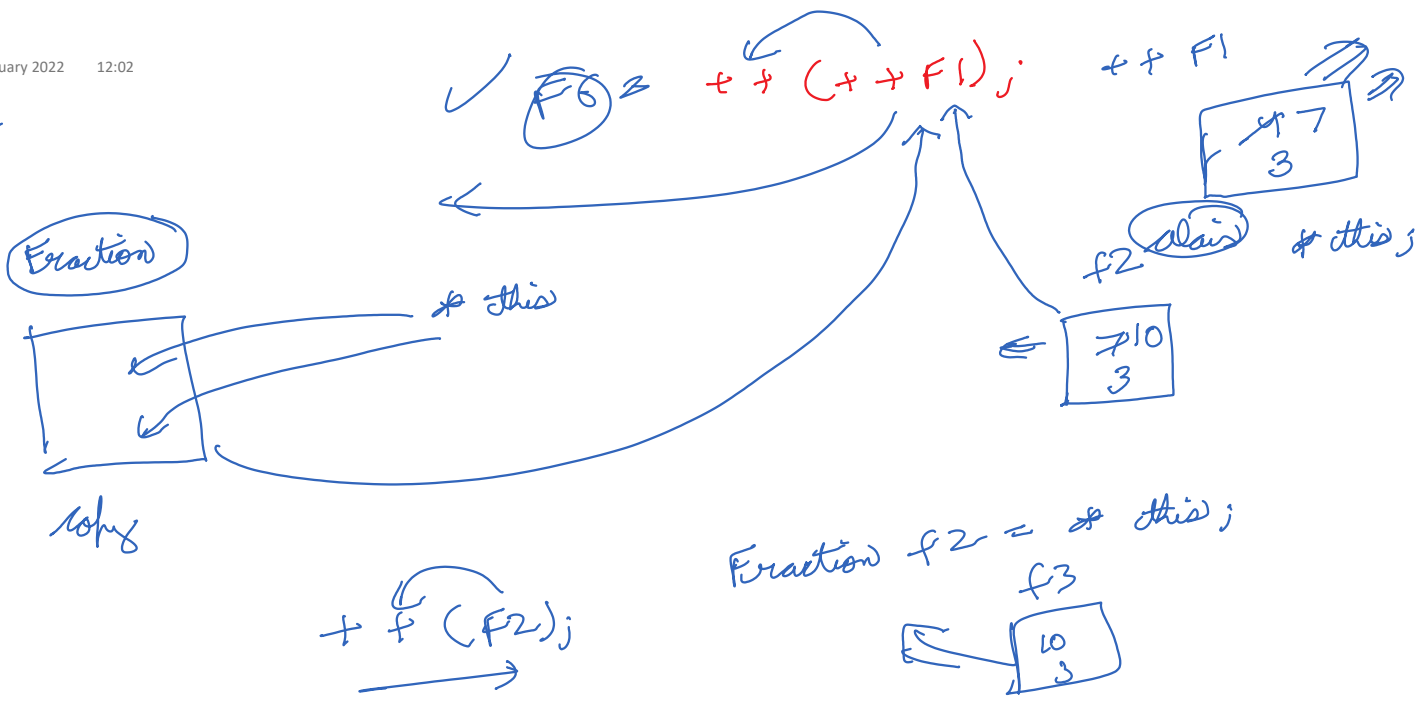
$\bar{w} = 7$
 $\bar{j} = 7$

F2

$= \textcircled{++f1} ; \Rightarrow f1.opato++()$
 \Downarrow
 $f1.opato++()$

F2 = F1 ;

*



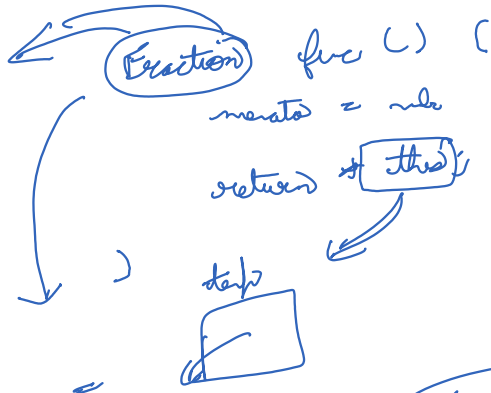
int w = 8; 6
int &k = w;

k & p;

Fraction

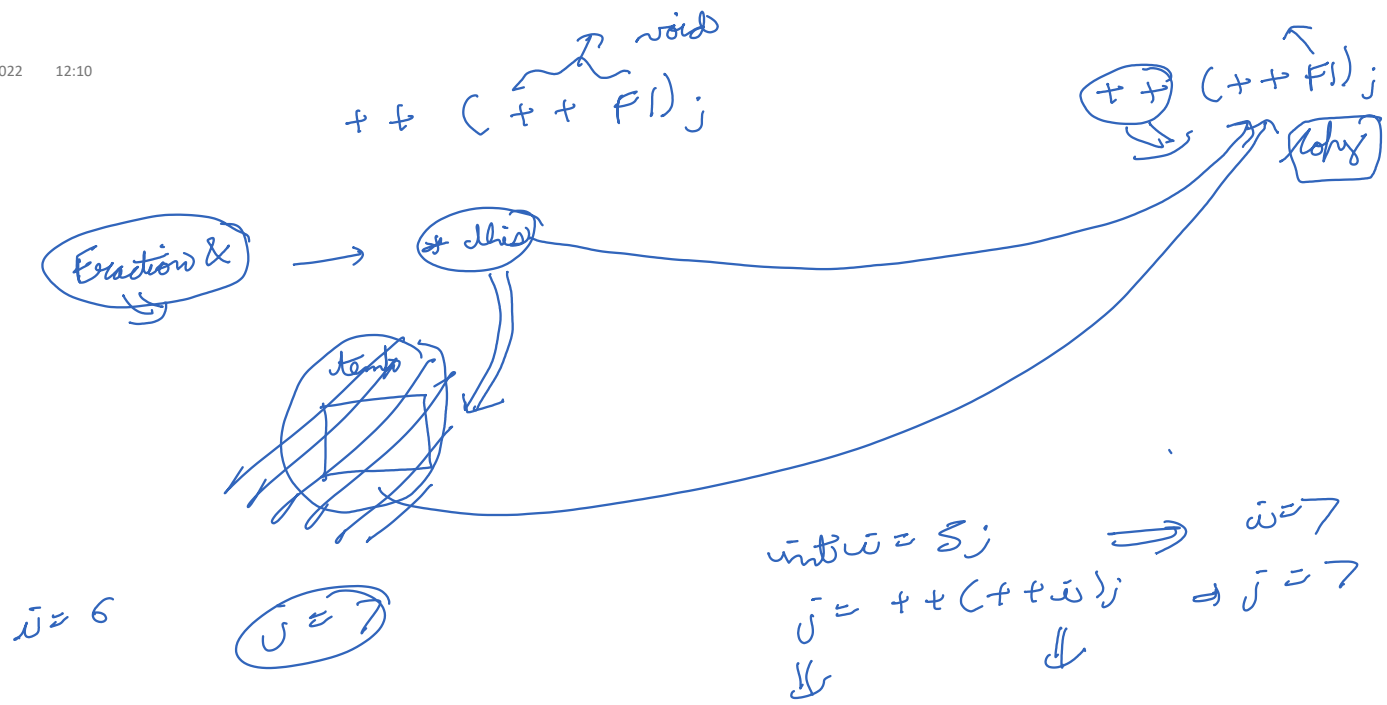
↑
this;

Fraction



Fraction f2 = new Fraction();
this

Fraction f2 = # this;



int &K = wj

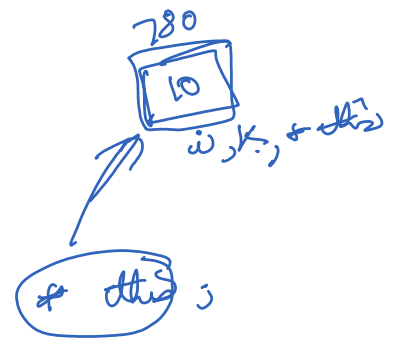
int wj = Kj

int @ = Sj

int j = ++(++w); k++;

++j;

int @ = 10j
int &K = wj



* this

Equation &

step f_1 ←
 $\begin{array}{|c|} \hline 37 \\ \hline 4 \\ \hline \end{array}$
 * this

Equation & denp =

Equation $f_6 = ++f_1j$

int $\bar{w} = S_j$
 int & $k = \bar{w}_j$

int $\bar{j} = k_j \Rightarrow \bar{j} = 5$

$\bar{j} ++j$

$$\text{int } j = w + + j$$

↓

$$j = s_j \rightarrow$$

$$w = 6_j$$

$$w = s_j$$

$$\frac{31}{12} + \frac{5}{4} = \frac{31 + 15}{12} = \frac{46}{12} = \frac{23}{6}$$

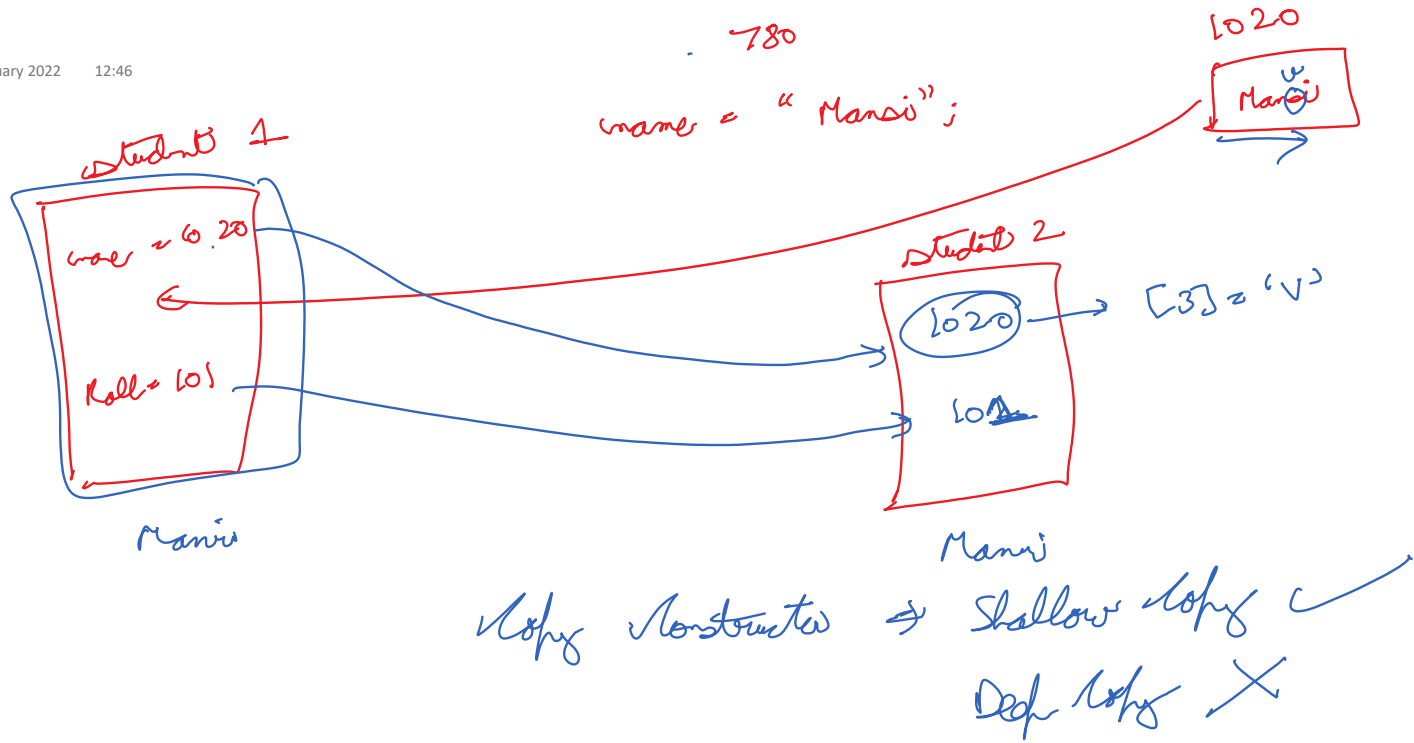
(w + +) + + j → X

$$F1 \oplus F2_j$$

$$\{ (f1 + = f2) + = f2 \}_j \text{ } f2 = f2_j$$

↓

Fracton &



student 2 (student 2 s1) {

}

student s1 (s2);

```
int arr[100];  
    ↳ Size (Fixed)
```

```
Capacity = 100 200  
total Elmts = 100 101  
  
int * arr;
```

size = ~~100~~ 200
total = ~~100~~ 101

Dynamic Array

```
Array ⇒  
total Elmts ⇒  
Capacity ⇒
```

1) Default constructor \Rightarrow capacity = 5
 total Elts = 0
~~arr~~ arr = new int [5];

2) copy constructor

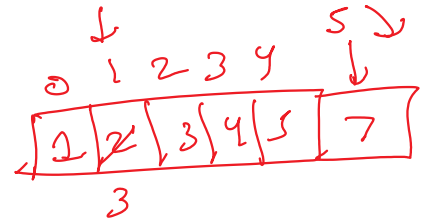
3) Operator = (copy Assigns Operator)

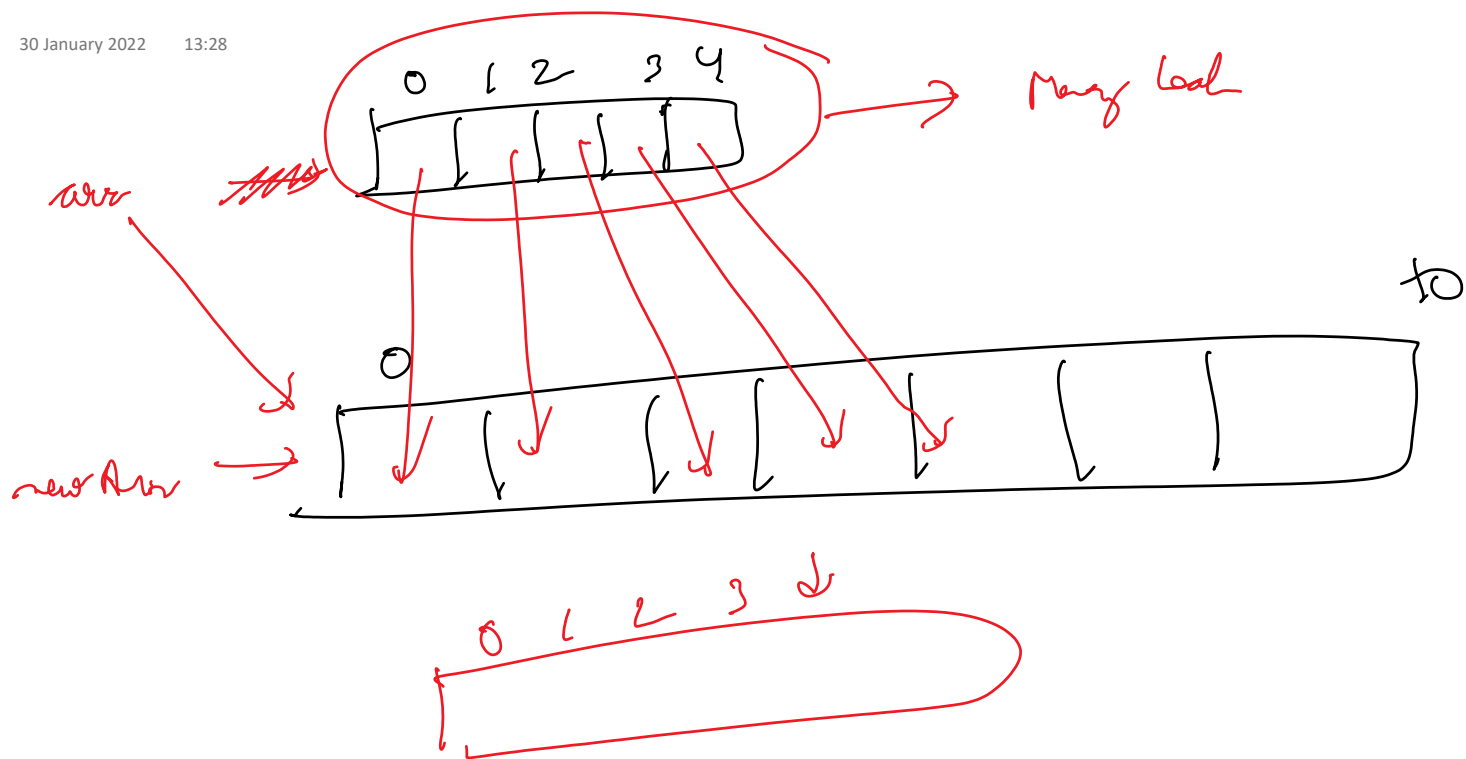
4) add (int) \Rightarrow $\left[\begin{array}{l} \text{capacity} = 100 \\ \text{total Elts} = 100 \end{array} \right] \xrightarrow{\times 2} \text{capacity} = 200$

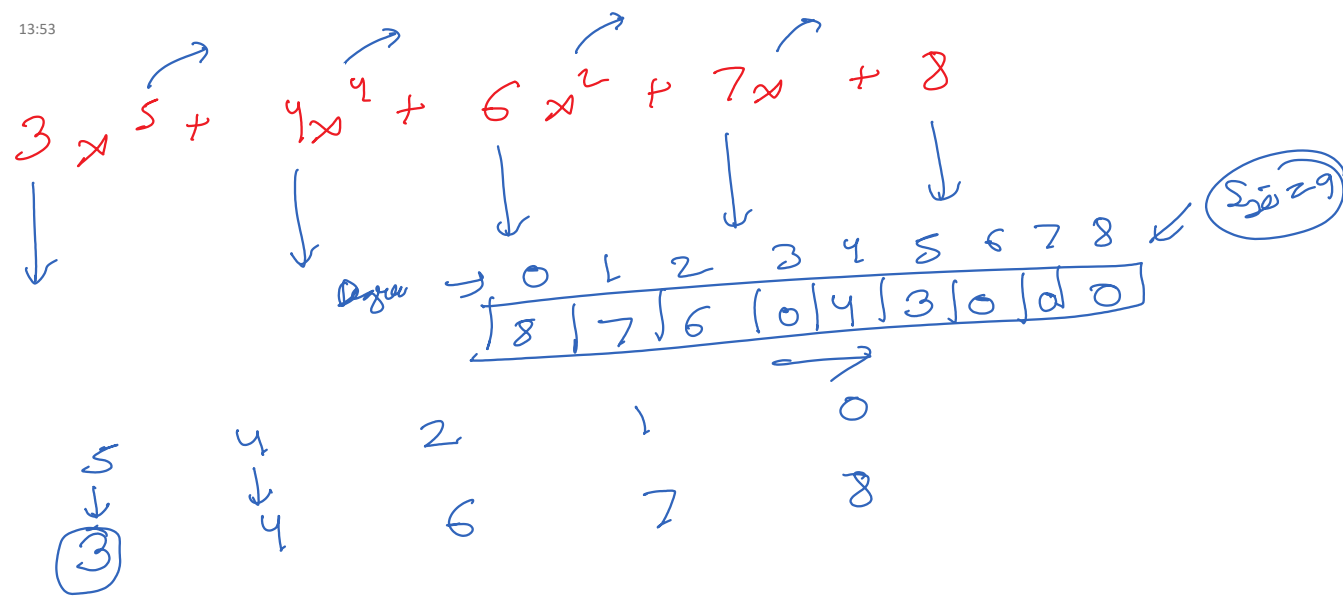
5) get (int i) \Rightarrow

6) add (int i, int elts) {
 }
 }

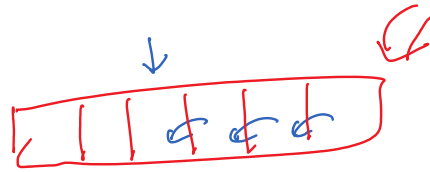
7) print()



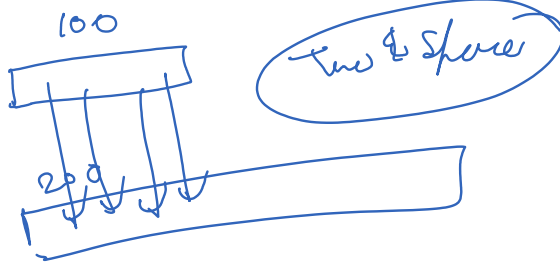




arr
arr²
arr³



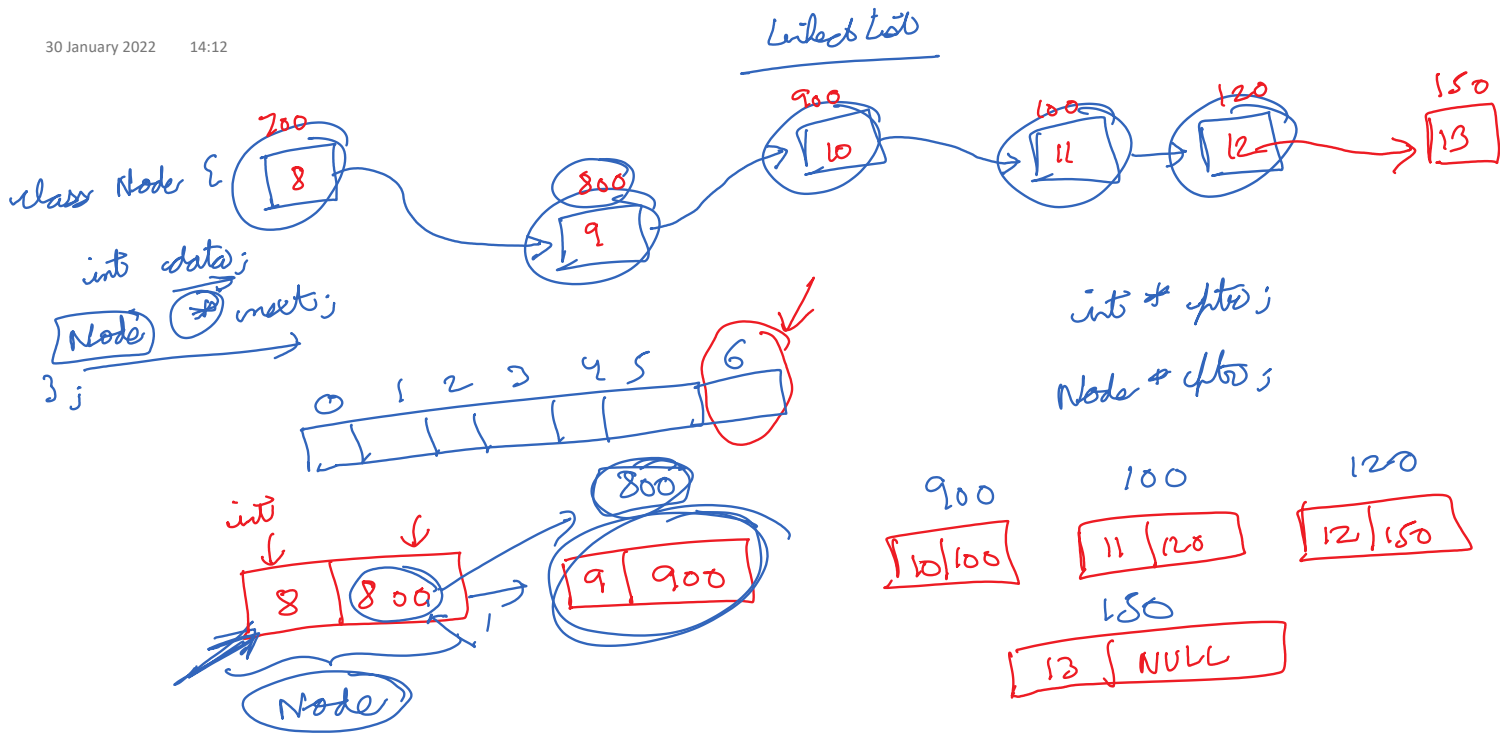
Array → Size

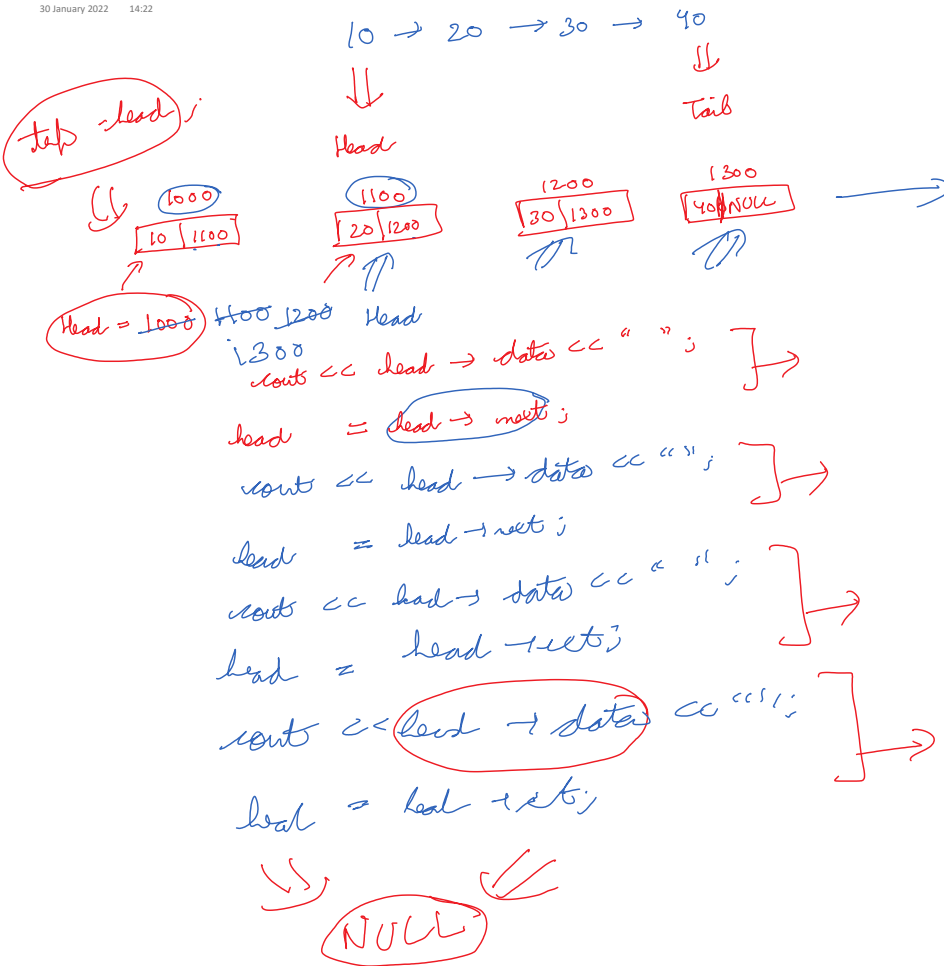


Data Structures

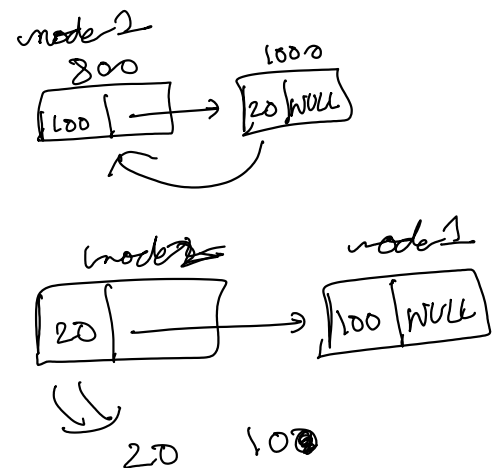
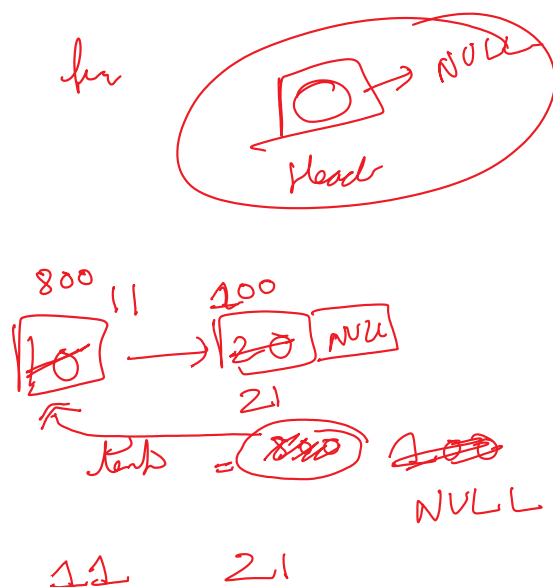
↳ Data Store

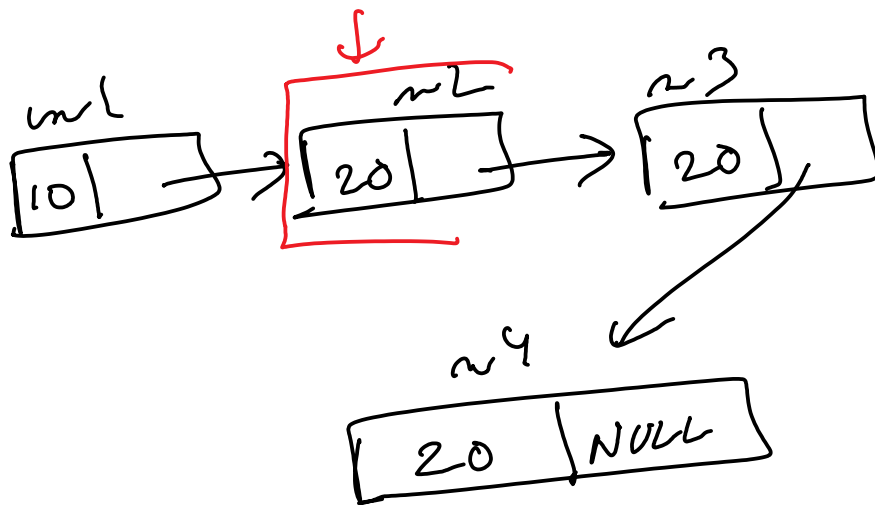
↳ Linear → Array





```
while (head != NULL) {
    next = head -> next;
    head = head -> next;
}
```





20 20 20