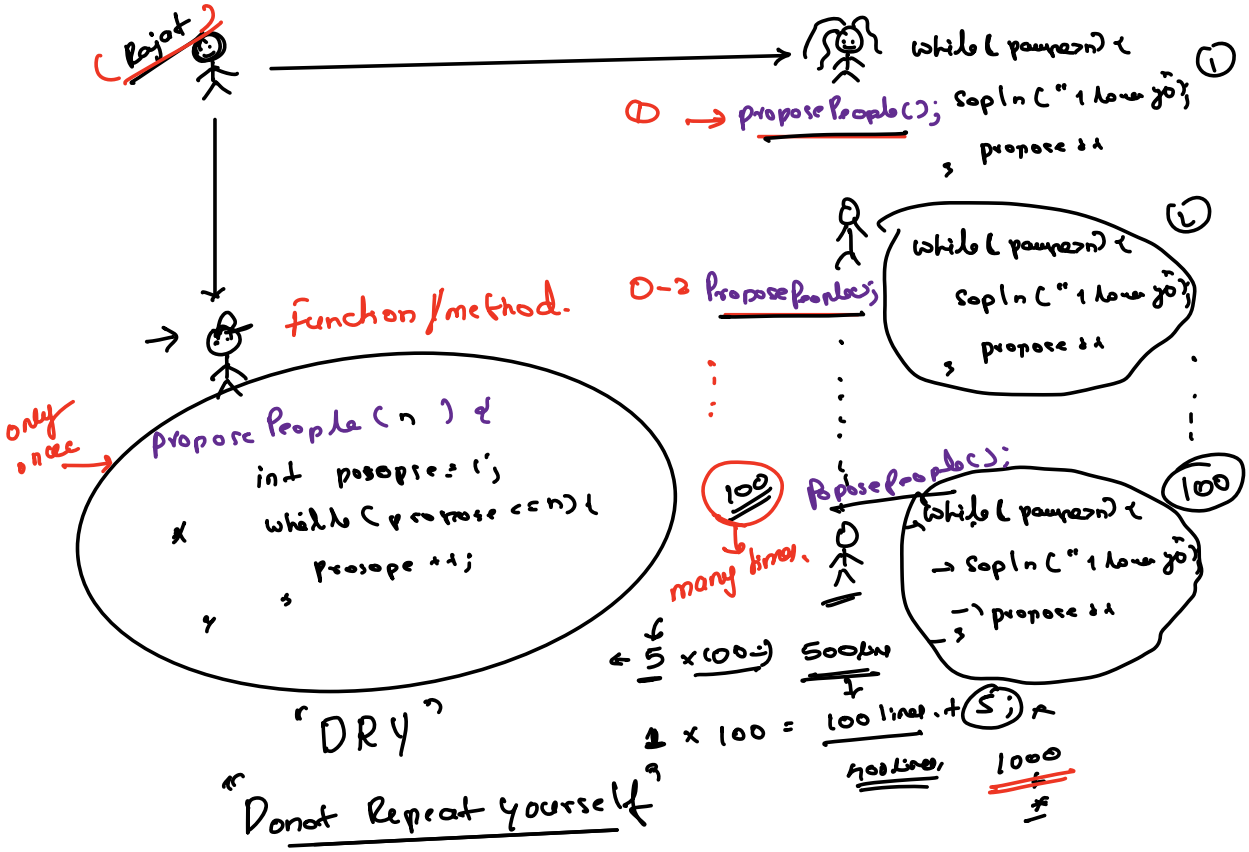


Q) Given n and r. ^nC_r

$$^nC_r = \frac{n!}{(n-r)! * r!} \Rightarrow \frac{4!}{(4-2)! * 2!} = \frac{24}{2 * 2} = \frac{24 \times 6}{2 * 2}$$

n = 4, r = 2 → 4 × 3 × 2 + 1 = 24, 2 × 1 = 2

i/p n = 4 and r = 2 ⇒ o/p → = 6 //.

$n! = 5 \times 4 \times 3 \times 2 \times 1;$

$\Rightarrow 1 \times 2 \times 3 \times 4 \times 5$ (5 = n)

① n! → fact1

② (n-r)! → fact2

③ r! → fact3

ans = fact1 / (fact2 * fact3)

```
int fact = 1;
for(int i = 1; i <= n; i++) {
    fact = fact * i;
}
```

factorial

without function

```
// n!
int nfact = 1
for(int i=1; i<=n; i++){
    nfact = nfact * i;        → (1)
}

// r!
int rfact = 1
for(int i=1; i<=r; i++){
    rfact = rfact * i;        → (2)
}

//(n-r)!
int nrfact = 1
for(int i=1; i<=(n-r); i++){
    nrfact = nrfact * i;      → (3)
}

int ans = nfact/(rfact * nrfact)

sop(ans);
```

with function.

```
int fact(int n){
    int fact = 1
    for(int i=1; i<=n; i++){
        fact = fact * i;
    }
    return fact;
}
```

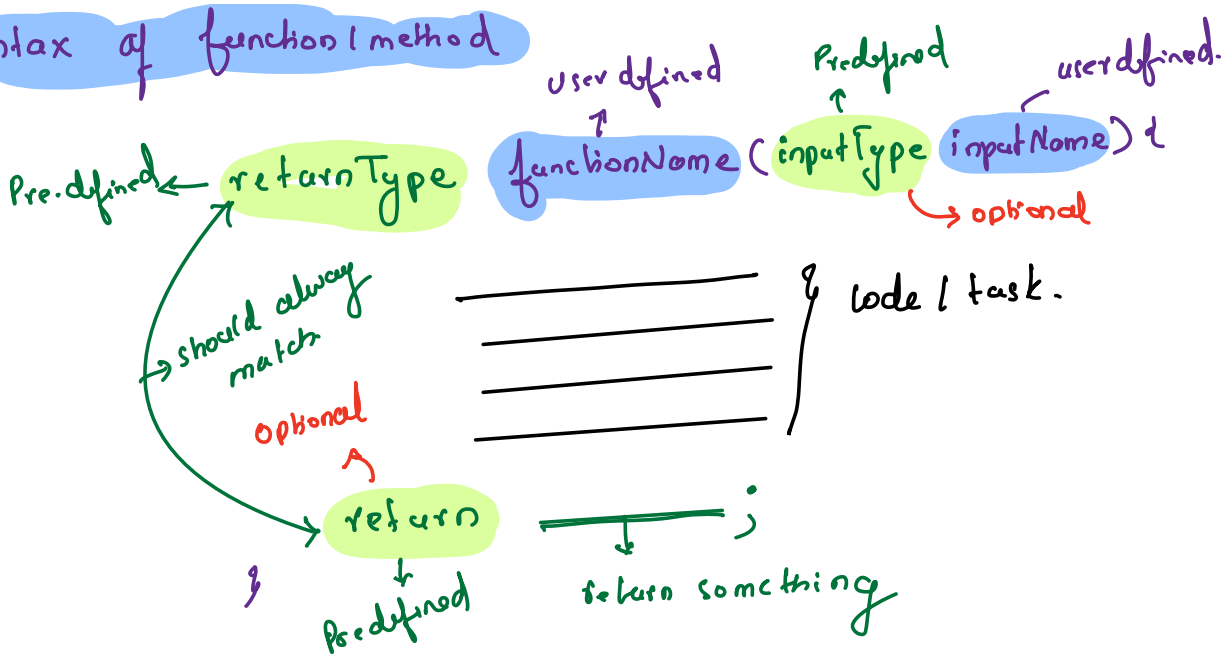
only once

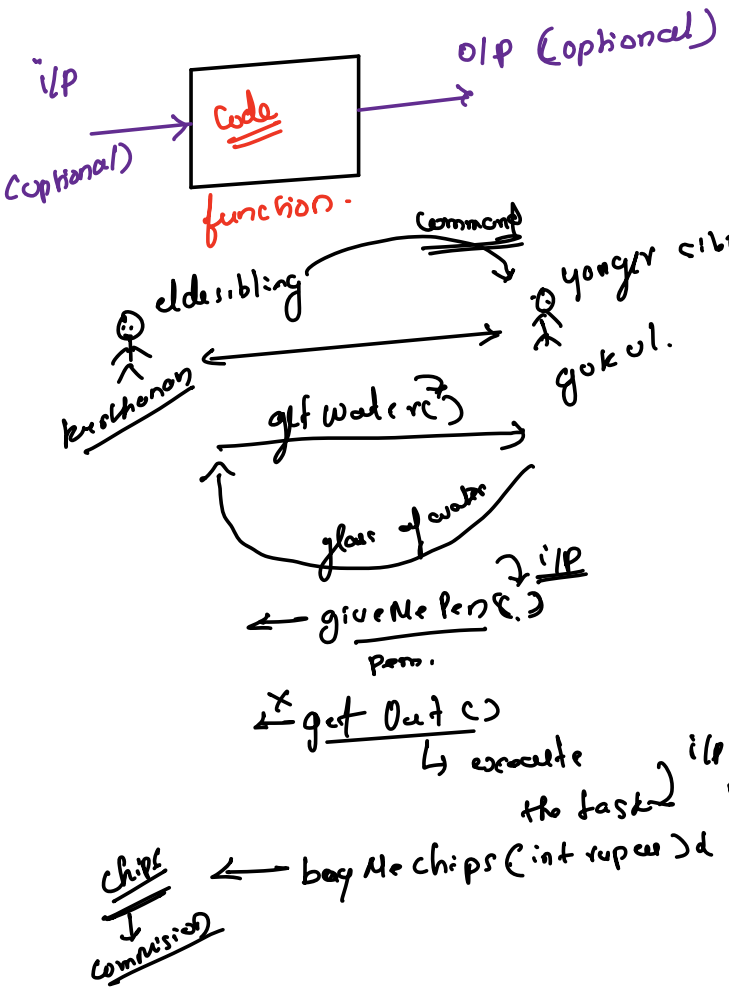
```
int nfact = fact(n);      → (1)
int rfact = fact(r);      → (2)
int nrfact = fact(n-r);   → (3)
int ans = nfact
          (nrfact * rfact)
sop(ans);
```

issues / Problems

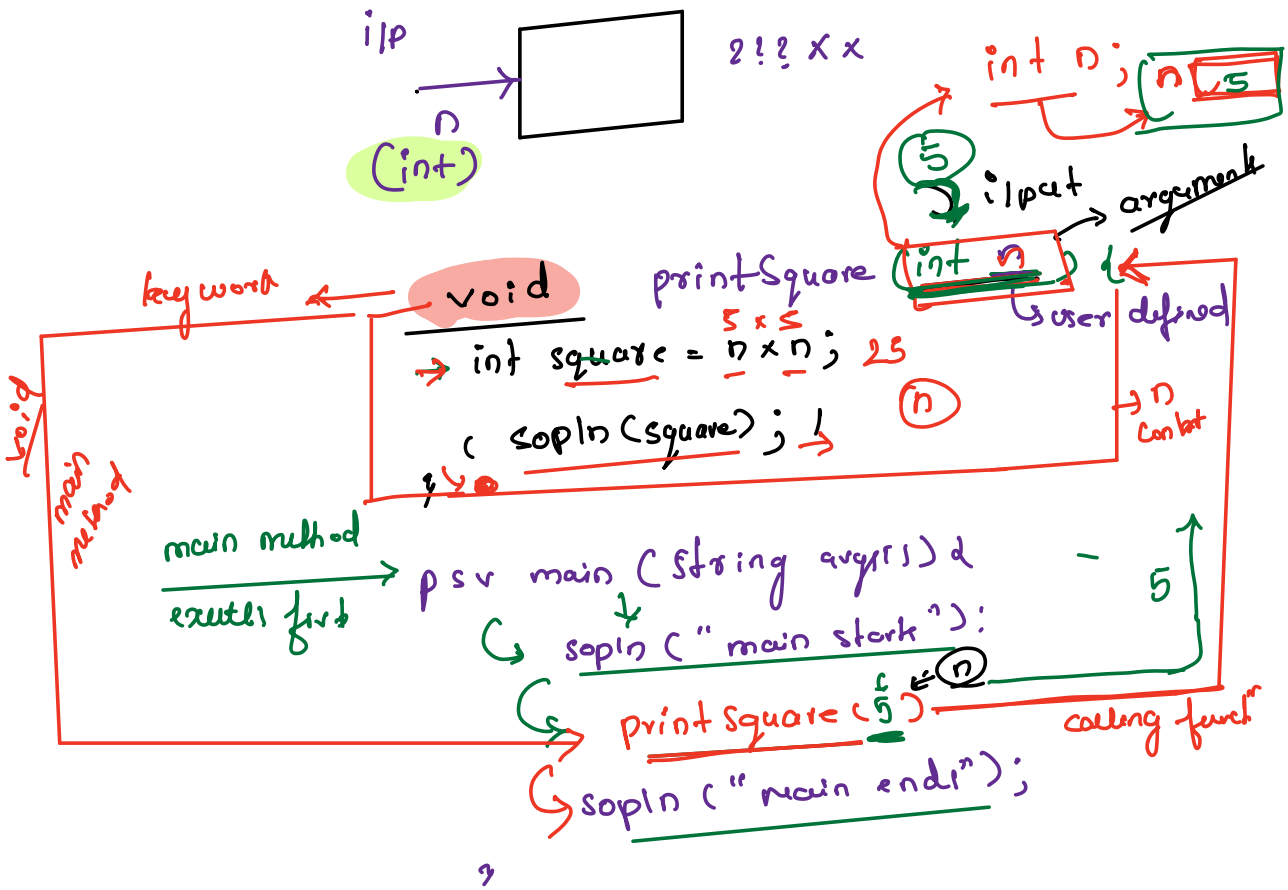
- → code repetation.
- → lengthy / time consuming → Functions / methods
- → Error prone.
- → Maintaince hard.
- → readability is bad

Syntax of function / method





Q) Given n write a function, that prints square of n;



O/p) main starts.
25
main ends;

break();

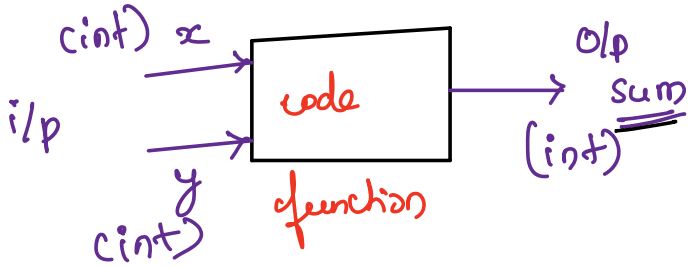
Sponsor "Rajat" = Back

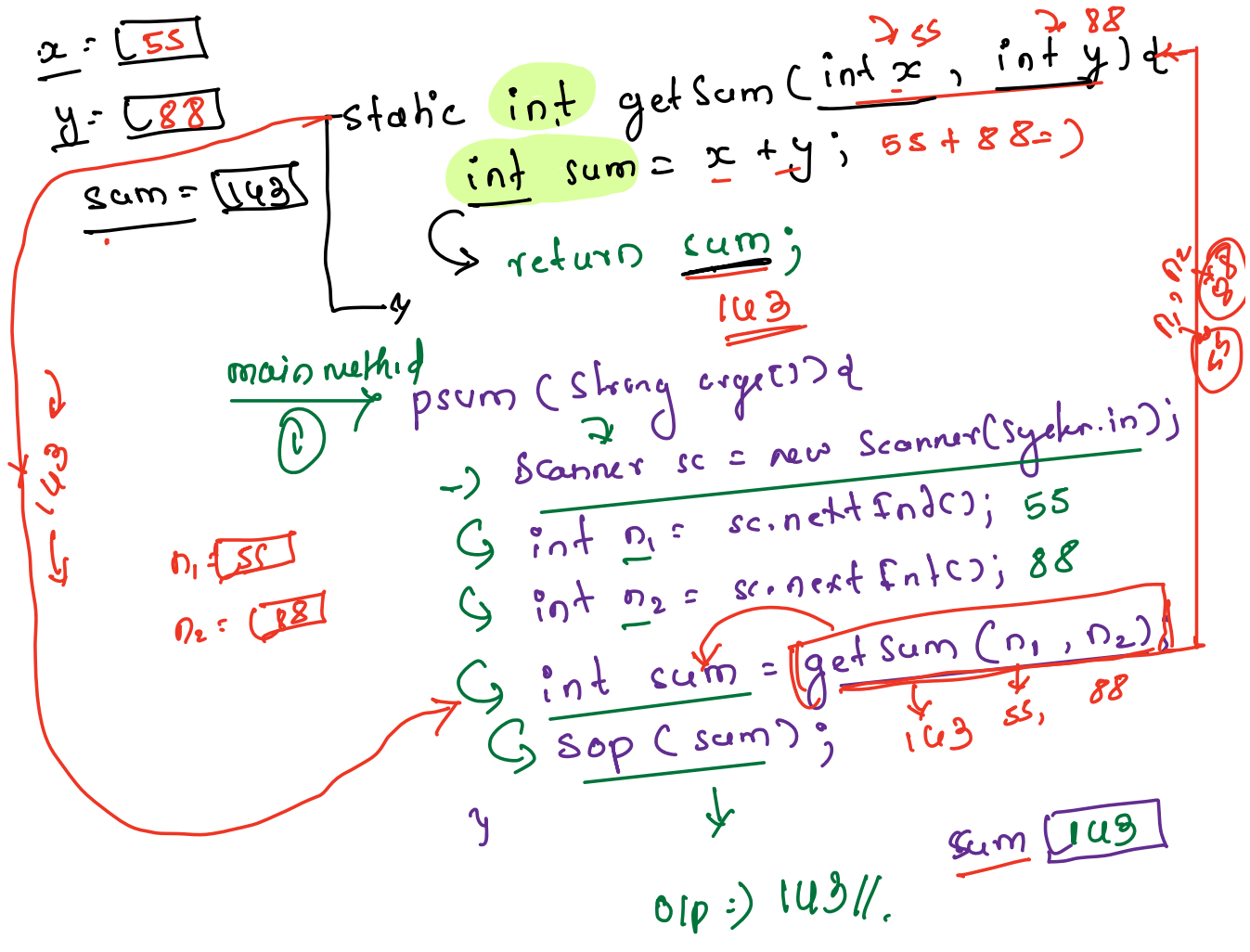
10:40 ⇒ 10:55 pm.

Rules of using a function/method.

1) function should be within the class and outside main method.
2) Main method/func will get executed first.
3) To use the funct^n/method we need to call it.
   functionName(i/p);
4) call the function inside main method.
5) static keyword before returnType.
6) void has no return statement.
7) we can call function "many time";
8) when we call the funct^n execution will go to that function and it will return to the main method.
9) returnType and return ——;
   should be same;

Q. Given two input x and y return the sum.





Quiz 1//

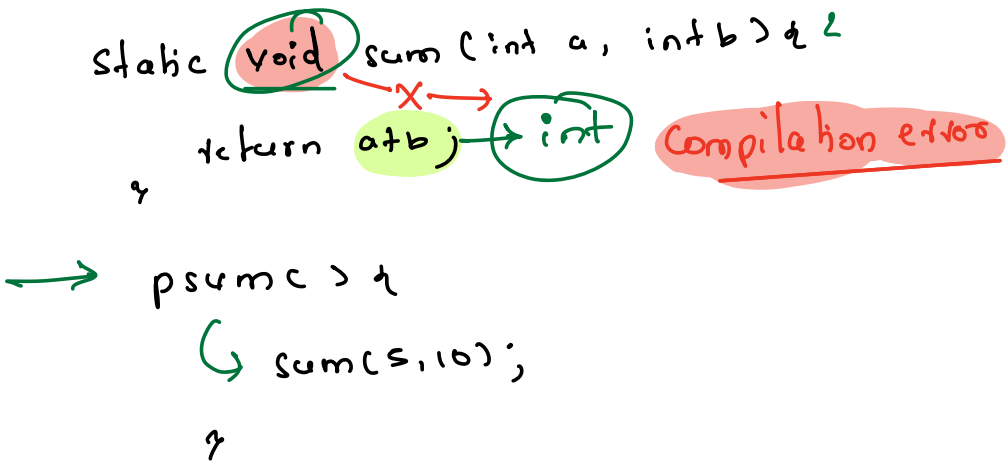
```
static int sum(int a, int b){
    return a+b;
}
psum(){
    sum(5,10);
}
```

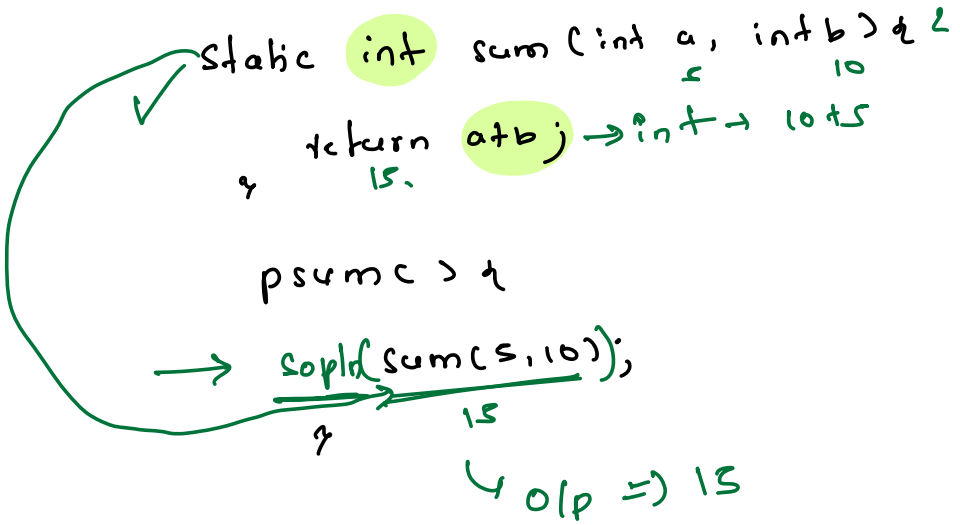
a=5 b=10

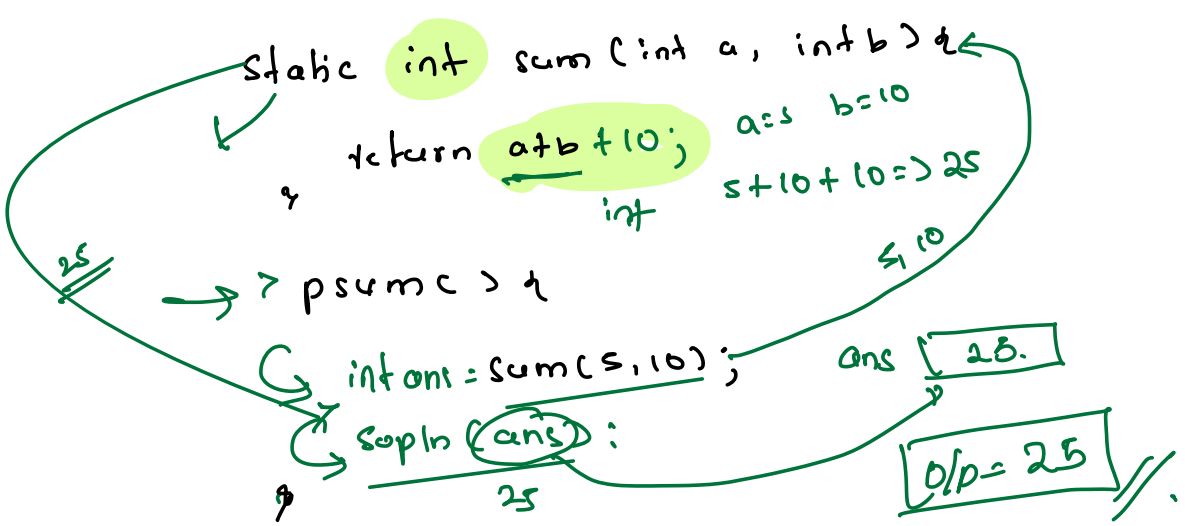
5+10 => 15

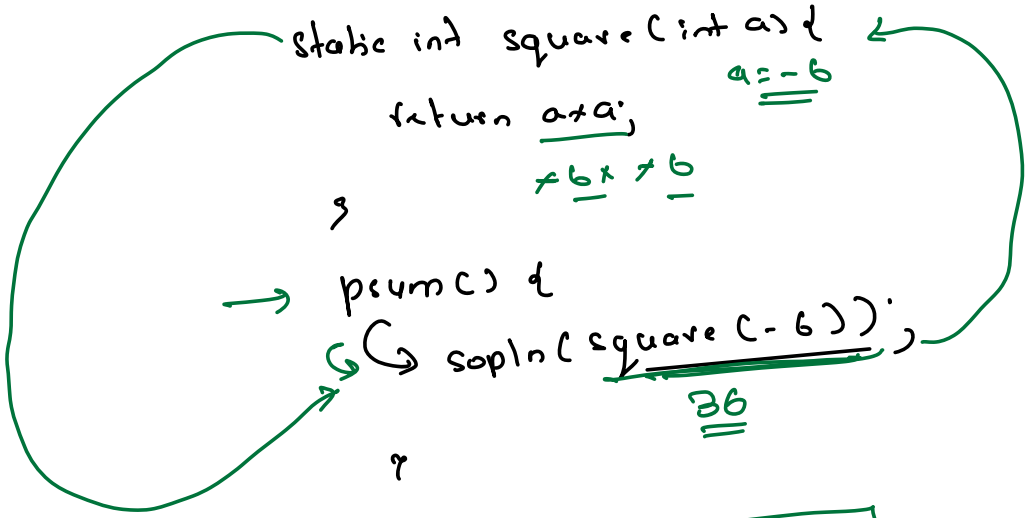
15

None//

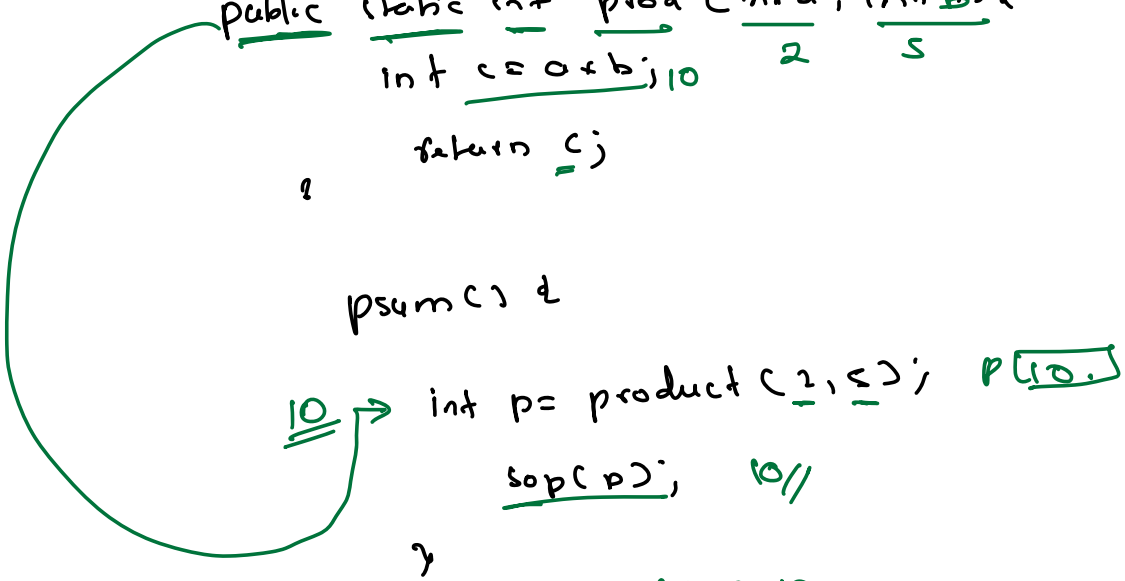

static void sum(int a, int b){
return a+b;
int
Compilation error
psvm(){
sum(5,10);


static int sum(int a, int b){
5
10
return a+b; → int → 10+5
15.
psvm(){
sopln(sum(5,10));
15
o/p => 15

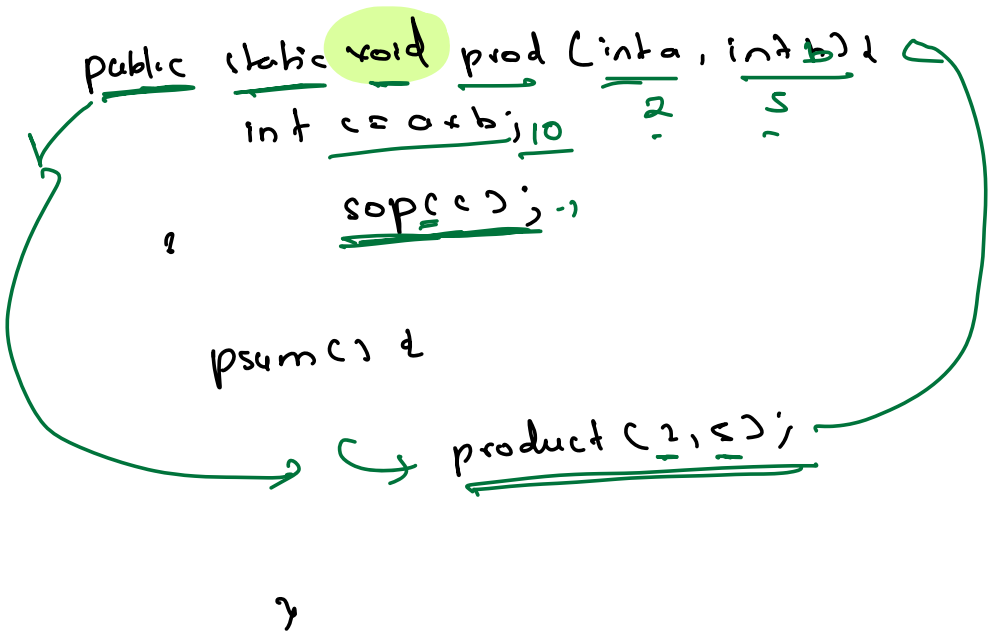

static int sum(int a, int b){
return a+b+10;
int
a=5 b=10
5+10+10=> 25
25
10
psvm(){
int ans = sum(5,10);
sopln(ans);
25
ans 25.
o/p= 25


static int square(int a){
a=-6
return a*a;
-6*-6
}
psum(){
sopln(square(-6));
36
}

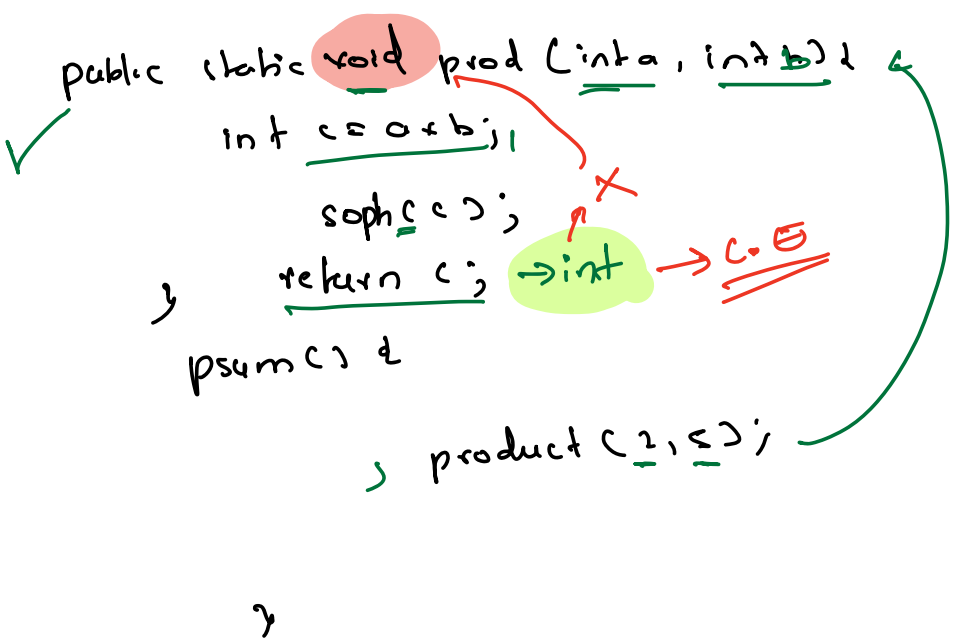
O/p => 36


public static int prod(int a, int b){
2
5
int c=a*b;10
return c;
}
psum(){
10
int p= product(2,5); p[10.
sop(p); 10//
}

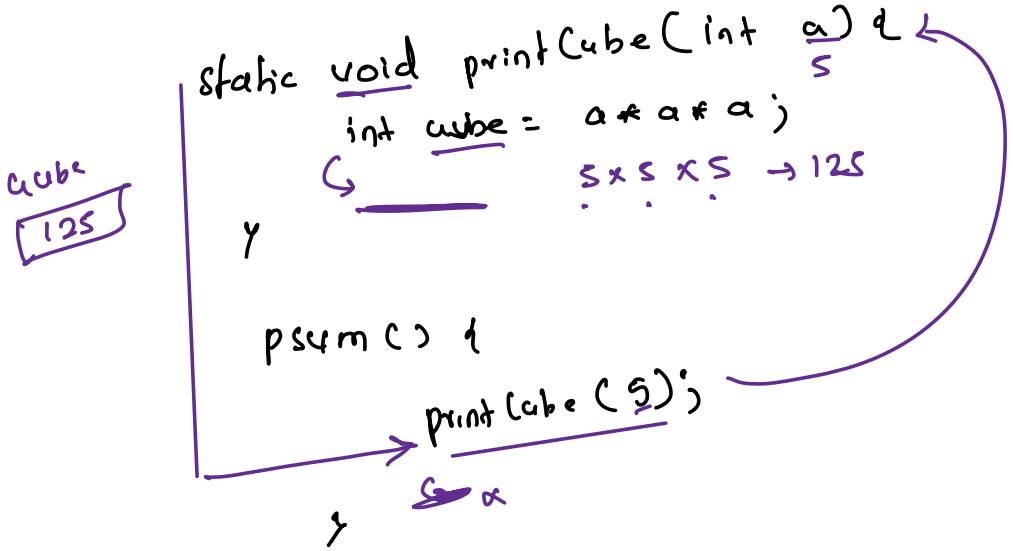
O/p => 10.


public static void prod (int a, int b) {
int c = a*b; 10
2
5
sop(c);
psum() {
product(2,5);
}

10


public static void prod (int a, int b) {
int c = a*b;
sop(c);
return c;
int
c.e
psum() {
product(2,5);
}

Doubt

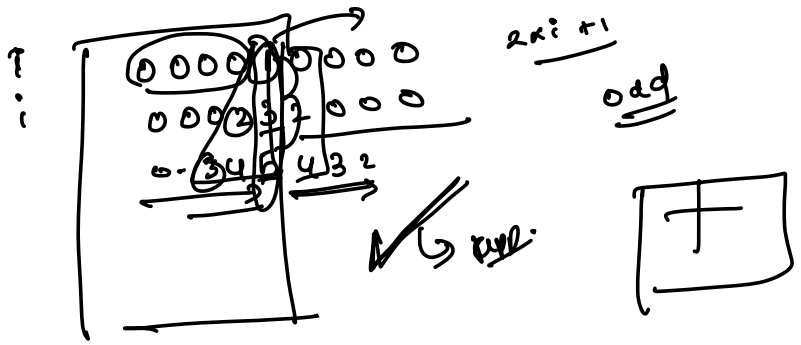


None

return type

```
public static boolean isHigh() {
    if(eyes == "red") {
        return true;
    } else {
        return false;
    }
}
```

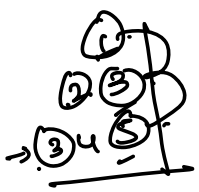
https://www.interviewbit.com/snippet/0668cd98b37166fc2da2/



for(zero)

for(increasing

for(decreasing



for zero