Quizz

```
static int sam(int a, int b) {
    int a = 20; b = 30;  |-> Compilation error:-
    return a+b;
}
```

Imp:- if variable name is already defined in input arguments. we cannot declare them again.

Quizz2

```
                         100      200
static int sum( int a, int b ){
                       a= ~~100~~, b= ~~200~~
                          20         30
    -> a=20
    -> b=30
       return a+b;
}
```

state

~~cm~~, c

cm2

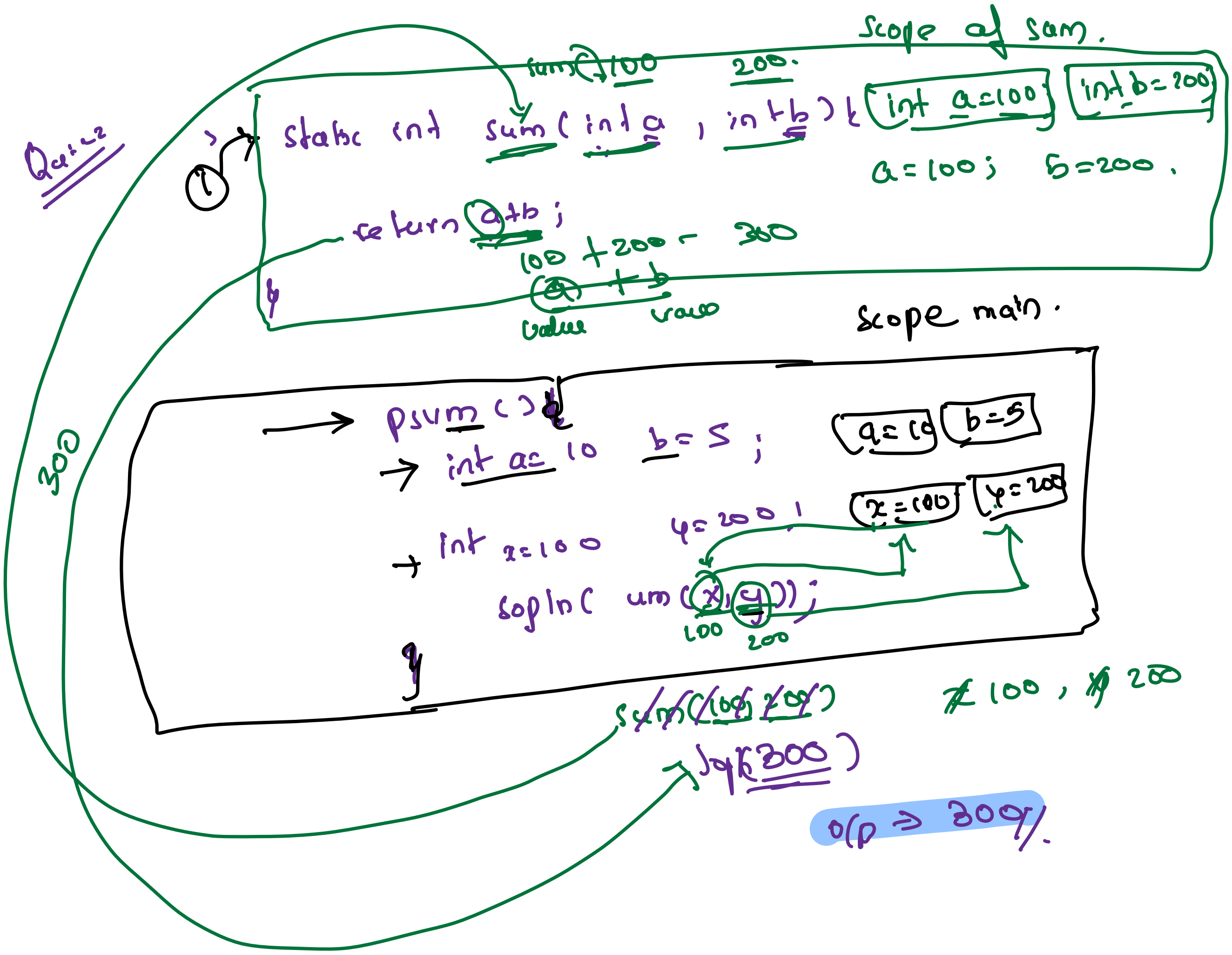
chief

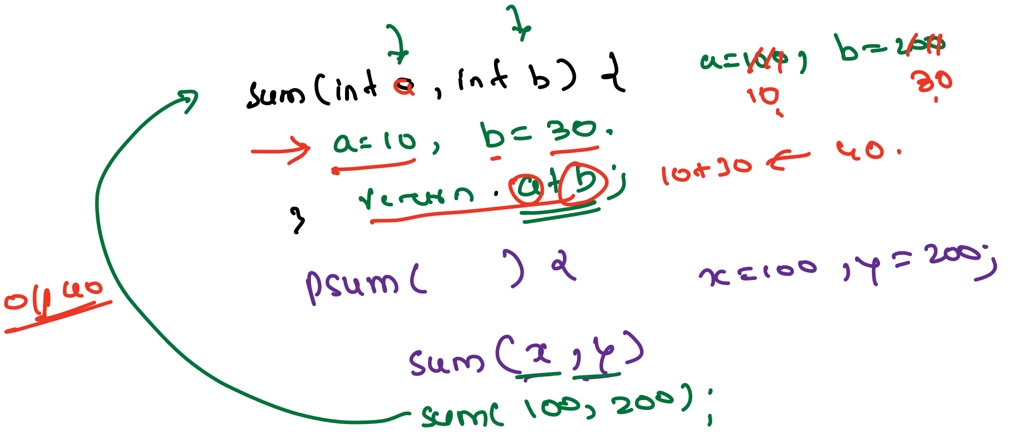
Imp:- We can reinitalize input variables / argument variables.

Quizz

```
①  static int sum( int a , int b ){           a=100 , b=200
        int x = 20  y = 30;                  x=20 , y=30.
        return a+b;                          100 + 200 = 300
    }
②

psvm(){
    int a= 10  b= 5 ;      a=10 , b=5
                                             } main
    int x=100  y= 200;     x=100, y=200
    sopln( sum(x,y));
}
        sum( 100, 200) ①
② 300
```


Quiz
Scope of sum.
sum(100 200.
static int sum(int a, int b){ int a=100 int b=200
a=100; b=200.
return a+b;
100+200 = 300
a + b
value value
}
1
Scope main.
psum() {
int a= 10 b=5;
a=10 b=5
int x=100 y=200;
x=100 y=200
sopln(sum(x,y));
100 200
}
x 100, y 200
sum(100 200)
Sop(300)
300
o/p => 300.


sum(int a, int b) {
a=~~100~~ 10, b=~~200~~ 30
a=10, b=30.
return. a+b;
10+30 ← 40.
}
psum( ) {
x=100, y=200;
sum(x, y)
sum(100, 200);
o/p 40

→ we cannot re-declare input variable. bcz they are already declared in scode of the method.

→ int variable scope is local to that method/ function

→ we can re-initialize the input variable. inside the method/ function scope.

→ when we call a function we are passing the values. / not variable..

ex. x=10 , y=20

sum(x,y) => sum(10,20);

→ when we return anything from a method we are only returning the value not the variable

a=10, b=30

ex: return a+b →

↓ return $\underbrace{10+20}_{30}$;

GCD → ? Greatest Common divisor

12	1	2	3	4	6	12		
24	1	2	3	4	6	8	12	24.

GCD – 12   a, b

1 → min(a, b);

Q. write a function that takes two nums and returns the GCD of two numbers.

```
int a= sc.nextInt();
int b= sc.nextInc();
 int min =0;
  if (a <b){
     min = a;
    }else{
     min = b  → getMin(_,_)
   }
 int ans =0;
 for (int i=1; i <=min ; i++){
    if(a%i==0 && b%i==0){
        ans= i
     }       ↳ getGCD(a,b);
  }
 sop(ans);
```

```
  T
a%i==0
  ||
b%i==0
  F
```

```
psvm( ){
   int a= sc.nextInt();
   int b= sc.nextInt();
   Sopln(getGCD(a,b));
}
static int getGCD(int a, int b){
  int min = getMin(a,b);
  int ans =0;
  for (int i=1; i <=min ; i++){
     if(a%i==0 && b%i==0){
        ans= i
      }
   }
  return ans;
}
static int getMin(int a, int b){
   int min =0;
   if (a <b){
      min = a;
    }else{
      min = b  →
    }
} return min;
```

```
→ psvm(  ){
①   int a= sc.nextInt();        → a= 12
    int b= sc.nextInt();        → b= 24
    sopln(getGCD(a,b)); ⑤ // calling the function → sopln(12);
                                                        O/P → ⑫
              ⑫      12,24
}
static int getGCD(int a, int b){      int a= 12  int b= 24.
                                      a=12   b=24   min= 12
    int min = getMin(a,b);            int min= 12;
                  12;    12,24 ④
    int ans =0;                       ans= ~~0~~ → ans = 12
②   for (int i=1; i<=min; i++){
        if(a%i==0 && b%i==0){
            ans= i;
        }
    }
              12
    return ans;
}
static int getMin(int a, int b){   a=12, b=24
    int min =0;                    min= ~~0~~; 12;
    if (a<b){
        min = a;
③   else{
        min = b
    }
} return min; → 12
```

During the interview

gcd →

x ~~Math.min(a,b)~~ x

Math.pow(a,b);

functions.

↳ ~~gst~~ x   break up values

→ perfect square return

square root   4 → 2

2×2

3×3

5 → -1.

$i * i = n$

return i

else return

-1

Method
function

V. Imp

public static void main (   ) {

non. SM X getSum();

static method ✓

static

non-static

}

→ ~~static~~ int getSum ( )

non-static

→ ~~X~~ boolean isEven ( ) {

2-3 months

syntax

}

static → static int gcd (___) {

nox

}

n static → int getMin (___) {

}

5-mothe

2 —— 3 or 4 problem solving 2

syntax   javca very weel

2months

(pattern, grid, arry)   arry, aoya, 2d

string

Coding → pralle java

(Arry) ← if for

while

imposter syndrome →

3

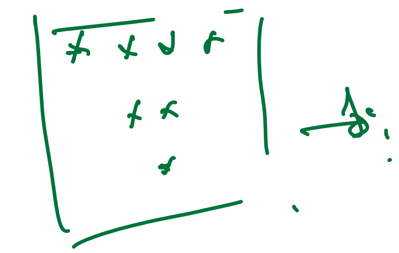
B com → Coding

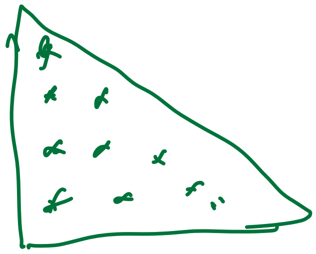
50 engin →

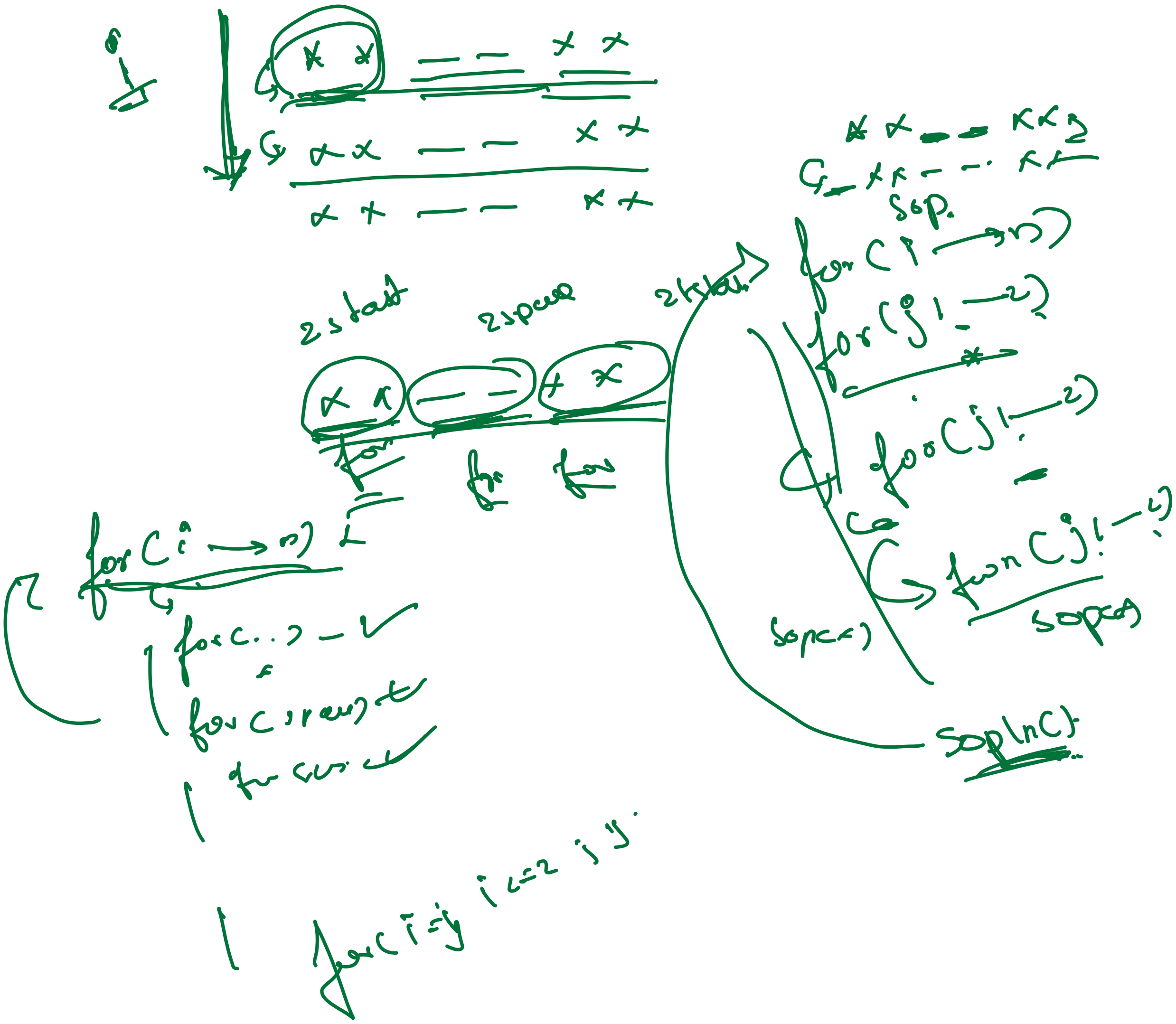
Apeus

Mechanical Eng → startups

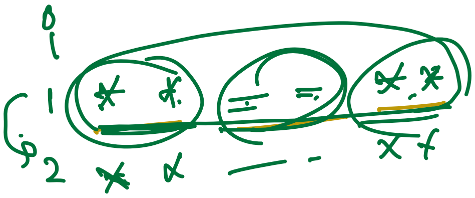
MNC,



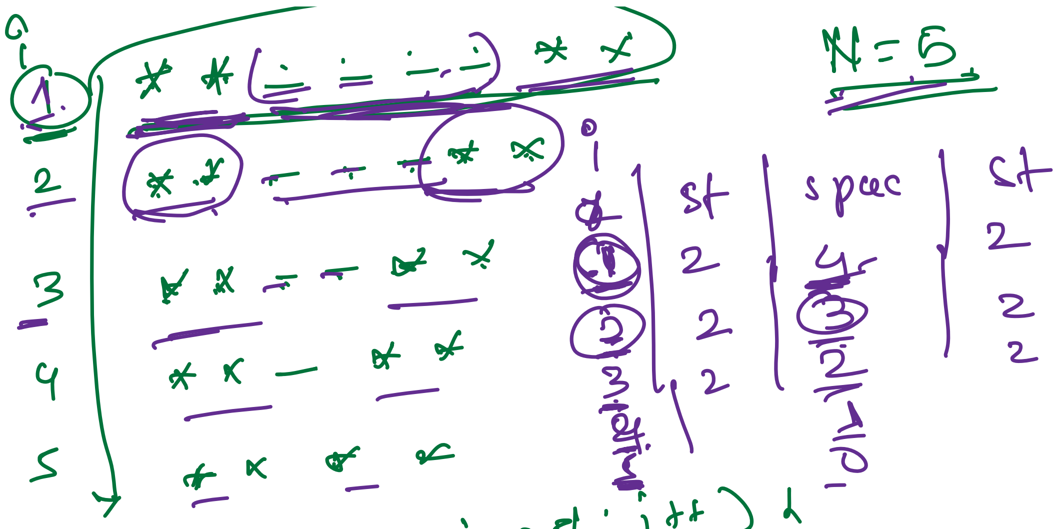




2 start
2 space
2 start
for
for
for
for(i →n)
for(...) —
for(space
for(i→n)
for(j! —2)
for(j! —2)
for(j! —2)
sop(...)
sop(n()
sop(...)
sop.
for(i=j i<=2 j,j.



```
for ( i=1 ; i <=2; i++)
        for (j=1 , <=2 j+d)
                sop(*)
        }
        for ( j=1 ; l=l j++)
                sop('—' );
        for ('j=1 : j l=1 j j++)
                sop(—);
```



```
for(int i = 1 ; i <= N ; i++) {
①→ for(i=1 ; j <= 2; i++){          2 2
        sop('*');
   }
   for(i=1 ; j <= N-i; j++) {        4 3
        sop("_ ");
   }
   for(i = 1 ; j <= 2 ; j++) {
        sop('*');                    2 2
   }
```

N = 5

N - i

i									
1	0	0	0	0	1	0	0	0	0
2	0	0	0	2	3	2	0	0	0
3	0	0	3	4	5	4	3	0	0
4	0	4	5	6	7	6	5	4	0
5	5	6	7	8	9	8	7	6	5

N=5

N=5

i

i	zero	nums	zero
N - 1	4	1	4
N - 2	3	3	3
N - 3	2	5	2
N - 4	1	7	1
N - 5	0	9	0

N - i

i - 1

i - 1

i * j

i N j

for (1 → N+i)

0

for (———)

nums

for (1 —— N - i)

0 .

```
static int sum(___) {
    if(___) {
        return ___;
    }
    return -1;
}
```

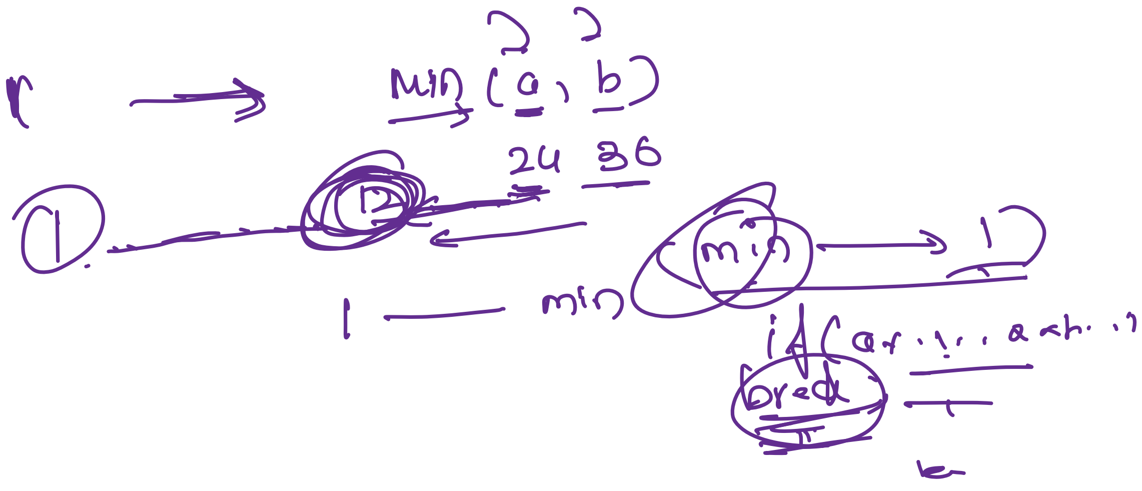
```
static int min(___) {
    if(___) {
        return ___;
    }
    else {
        return ___;
    }
}
```

java booleon int ,->?

true 0?

∞ int

false



https://www.interviewbit.com/snippet/25cfbb6f89fc5048063b/