CS & IT ENGINEERING

Data Structures

Introduction to Data Structures

Lecture No- 04

By- Pankaj Sharma sir

Introduction-3

Functions

printf
scanf

] pre-defined/Built-in function

① reusability

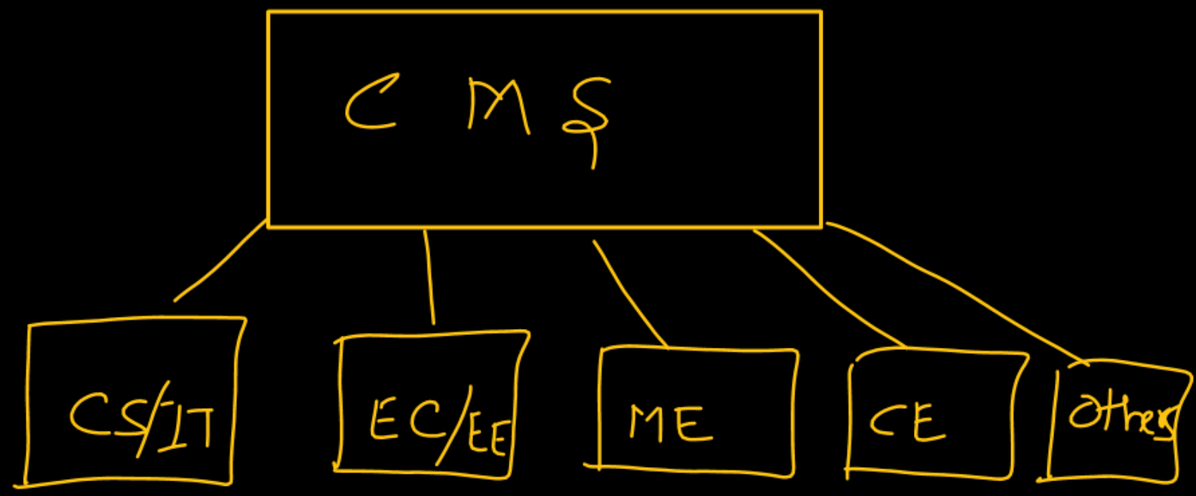
② Modular programming

1st way

```
Void main( ){

  1L lines of
  code

}
```



Function

ধারণা হচ্ছে

```
#include<stdio.h>

void main()
{
    int a=10,b=20,result;
    result = Satishsir(a,b);
    printf("%d",result);

}
```

a: 10

b: 20

200

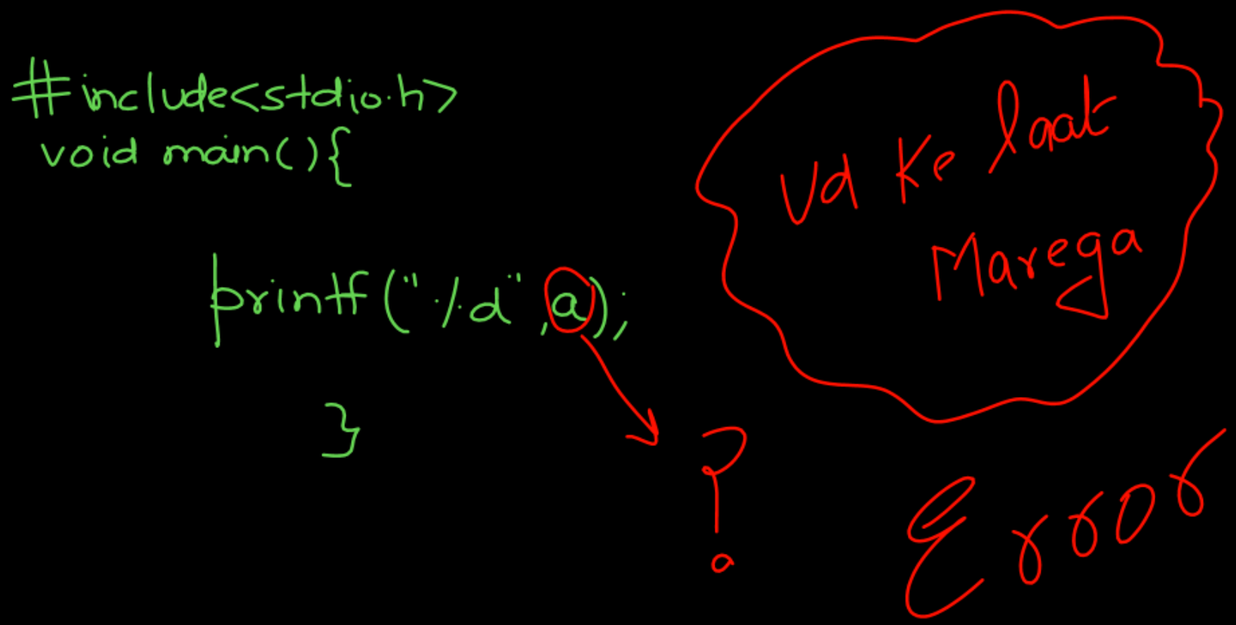
10 20

```
satishsir(int x,int y)
{
    int mult;
    mult = x*y;
    return mult;
}
```

x: 10

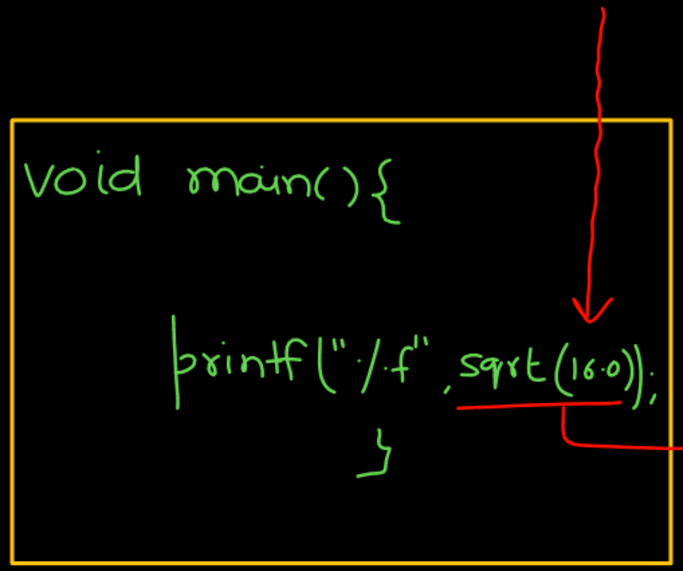
y: 20

mult: 200


#include<stdio.h>
void main(){
printf("%d",a);
}
Vd Ke load Marega
?
Error

Compilation

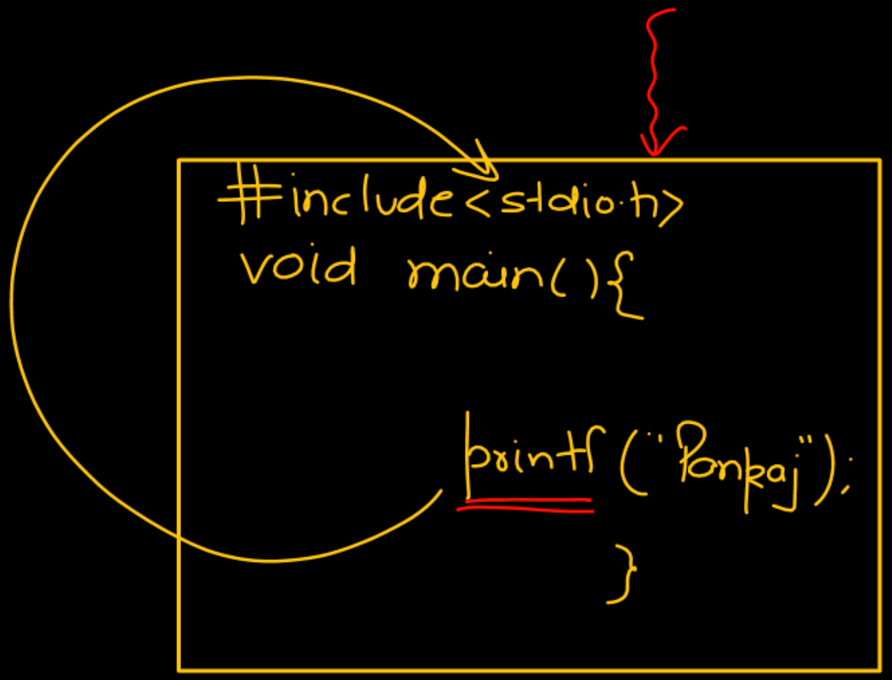
Execution

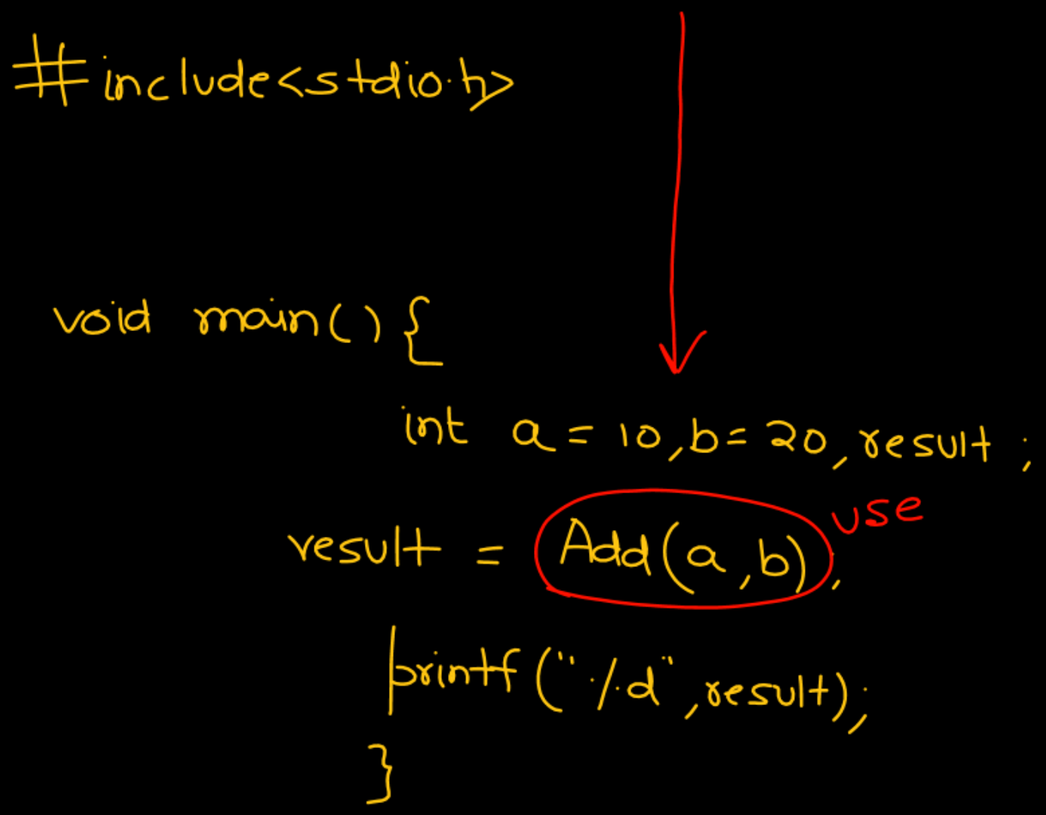


Use a function

?

What's Sol.


#include<stdio.h>
void main(){
printf("Pankaj");
}




```
int Add (int x, int y)
    {
        return x+y;
    }
```

```
#include<stdio.h>
int Add(int,int);
void main(){
    int a=10,b=20,result;
    result = Add(a,b);
    printf("%d",result);
}
```

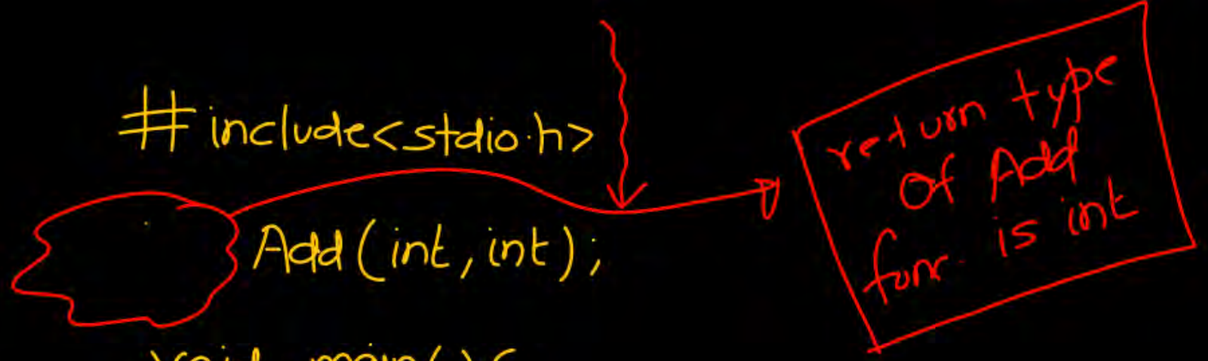
To avoid C.E
forward declaration

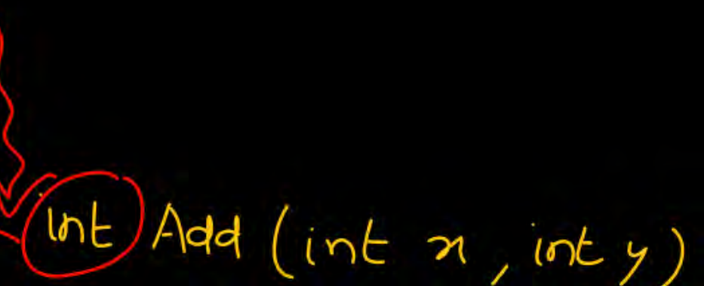
No Memory is allocated

```
int Add(int x, int y)
{
return x+y;
}
```

int Add(int ~~a~~, int ~~b~~);

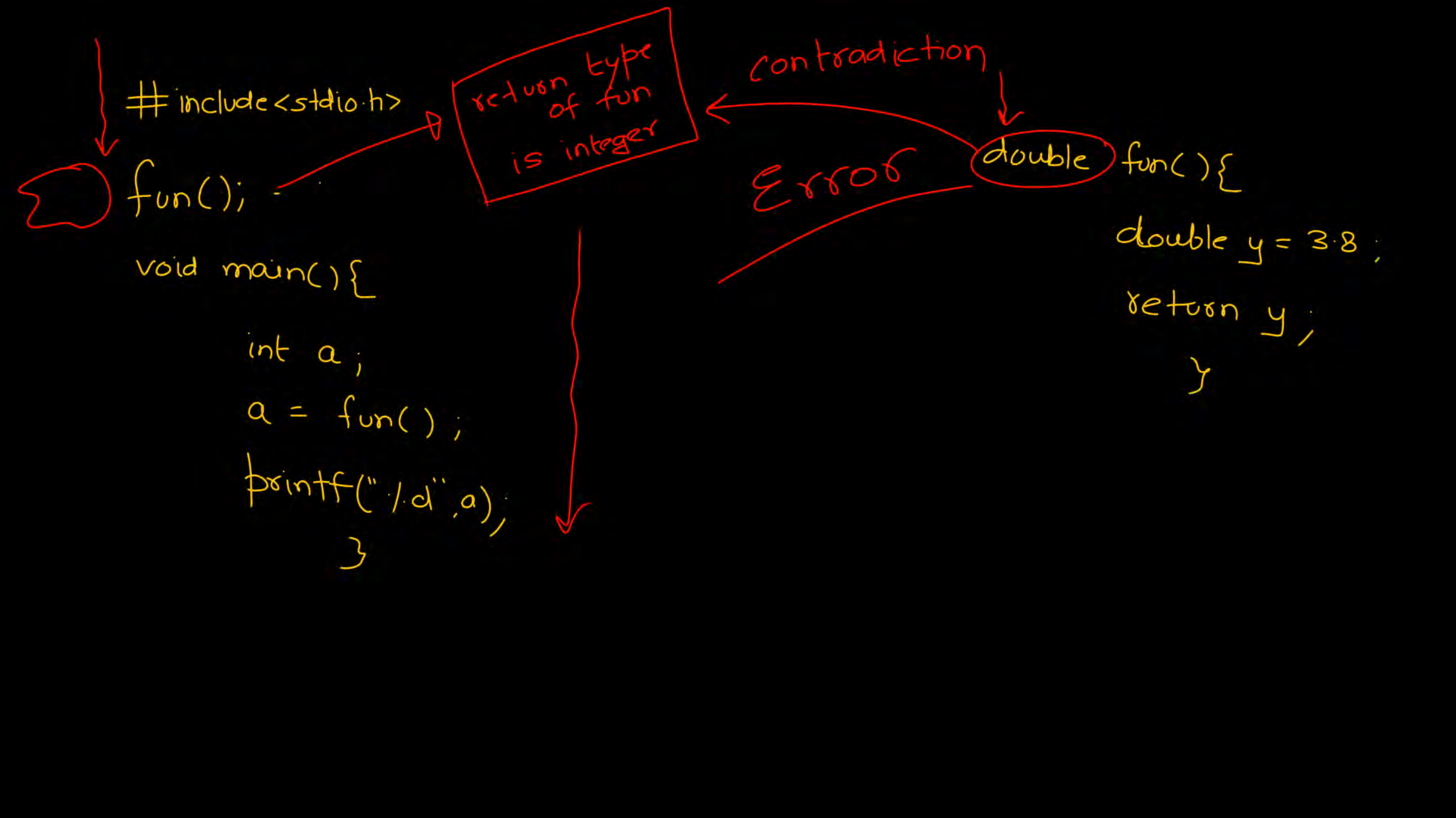
Useless





by default the return type of a function is integer

```
void main(){
int a=10,b=20,result;
result = Add(a,b);
printf("%d",result);
}
```


#include<stdio.h>
fun();
void main(){
int a;
a = fun();
printf("%d",a);
}
return type of fun is integer
contradiction
Error
double
fun(){
double y = 3.8;
return y;
}

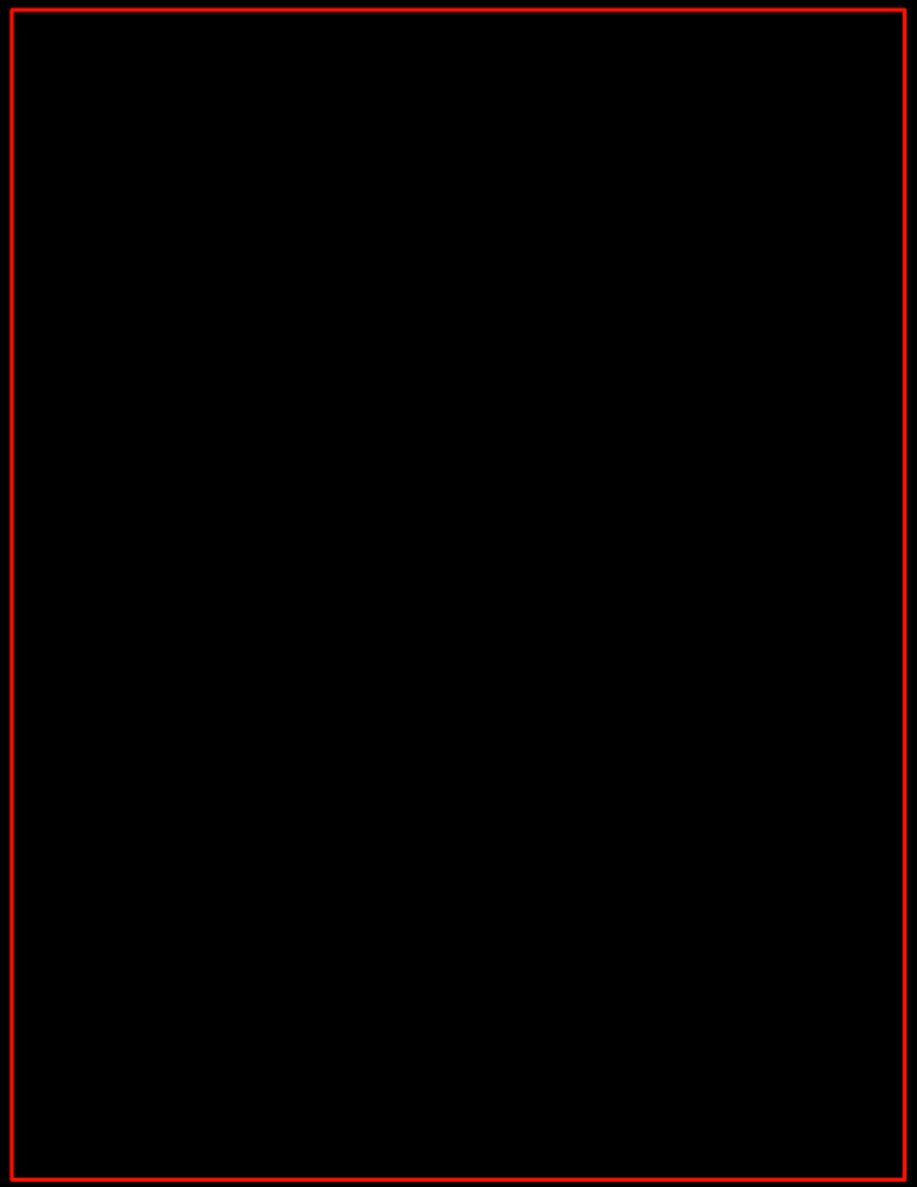
```
#include<stdio.h>

Add (int, int);
Void  main(){
        int a=10, b=20, resuH;
        result = Add(a,b);
        printf ("%d", result);
          }
```

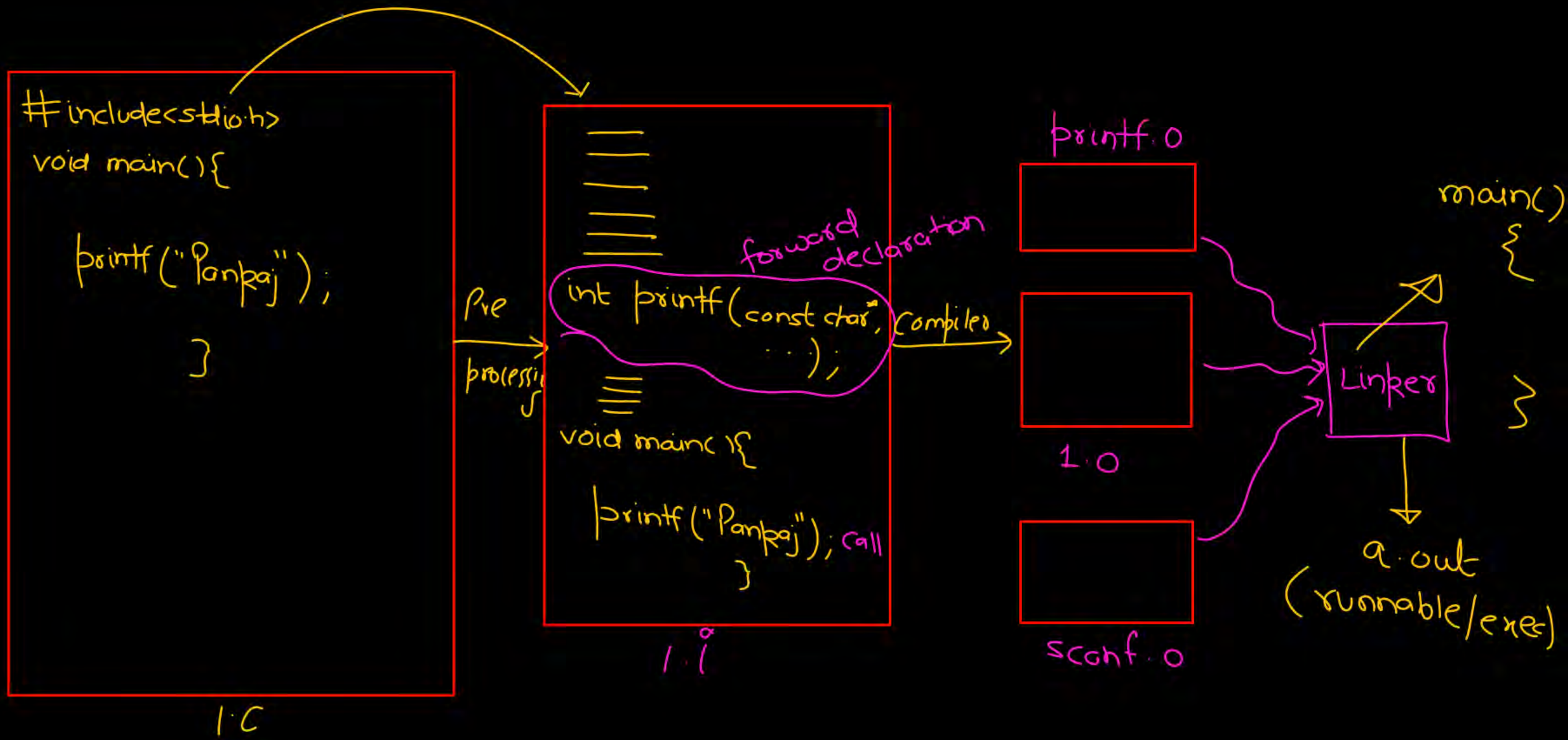
```
int Add (int x, int y)
  {
    return x+y;
  }
```

1. forward declaration
2. call/use
3. defin. /code

Header files



printf
scanf
} built-in
ये दोनों ही हैं


#includecstdio.h>
void main(){
printf("Pankaj");
}
1.C
Pre
processi
forward declaration
int printf(const char*, ...);
void main(){
printf("Pankaj"); call
}
1.i
Compiled
printf.o
1.o
sconf.o
Linker
main()
{
}
a.out
(runnable/exec)

```
#include<stdio.h>
int Add(int x, int y)
    {

     return x+y;
    }
```

Compile ✓

Execute ✗

Compiler → Linker

void → nothing Add

printf ⟹ integer value
↳ No. of symbols printed by it.

```
#include<stdio.h>
void main(){
    printf("Pankaj");
}
```

✓

```
#include<stdio.h>
void main(){
    int i;
    i = printf("Pankaj");
    printf("%d",i);
}
```

Pankaj6

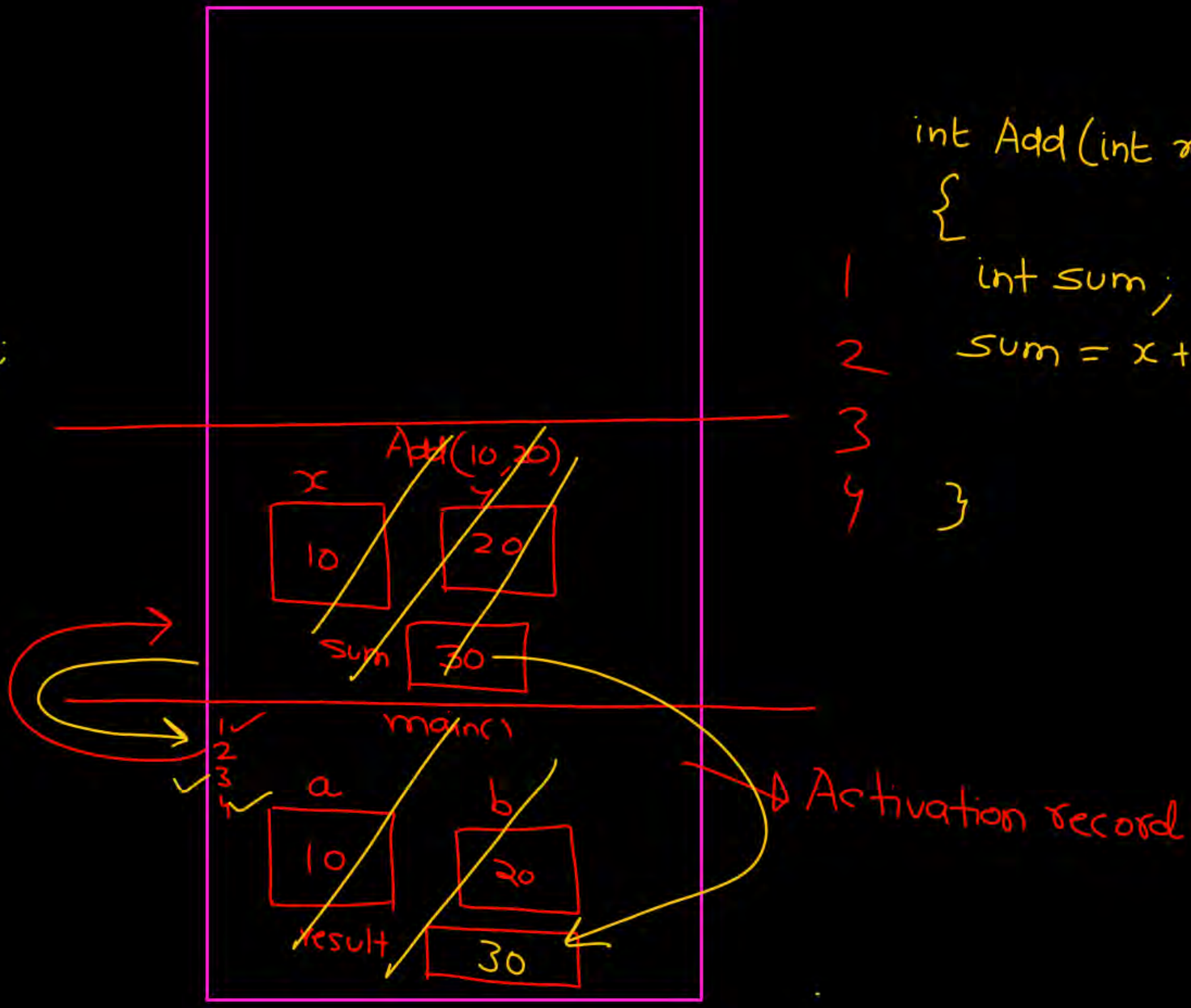
```
#include<stdio.h>
int Add(int,int);
void main(){
        int a=10,b=20;
        Add(a,b);
   }
```

```
int Add(int x,int y)
    {
int   sum;
     sum= x+y;

    }
```

```
#include<stdio.h>
int Add(int, int);
void main(){
1.  int a=10, b=20, result;
2.  result = Add(a,b);
3.  printf("%d", result);
4.          }
```

```
int Add(int x, int y)
{
1     int sum;
2     sum = x+y;
3
4   }
```



30

#

```
int Add(int, int);
void main(){
    int a=10,b=20, result;
    result = Add(a,b);
    printf("%d",result);
}
```

Calling function

10 20

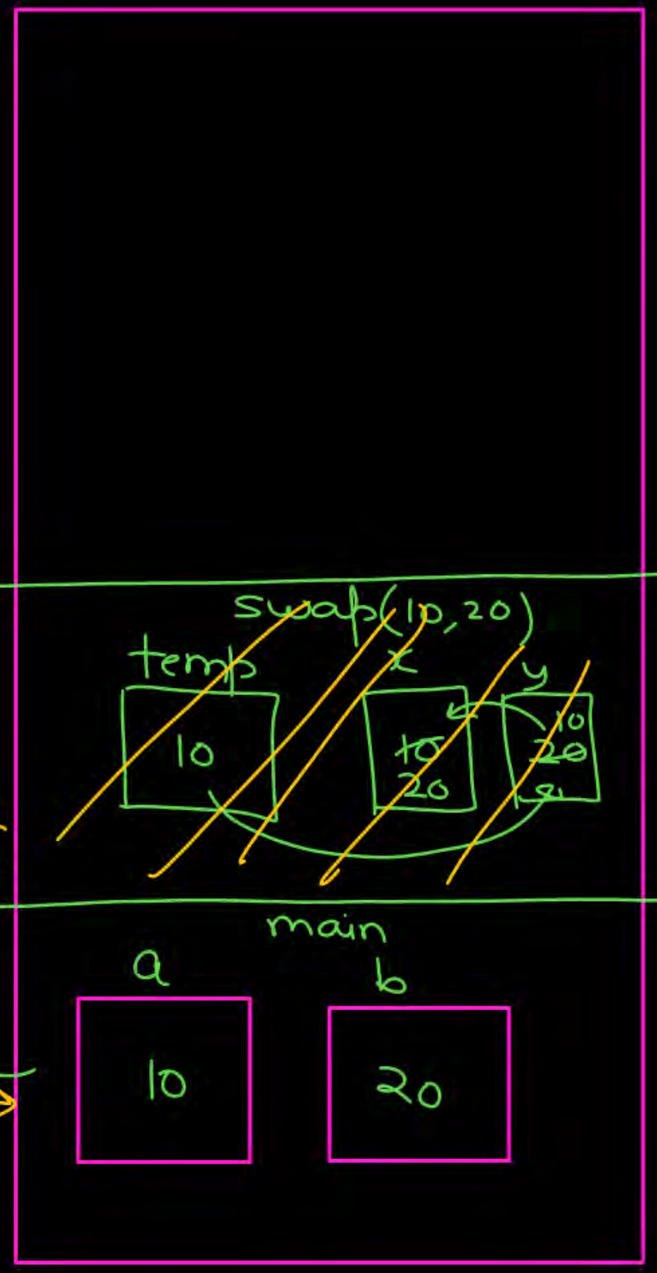
Actual arguments/
Actual parameters

formal argument

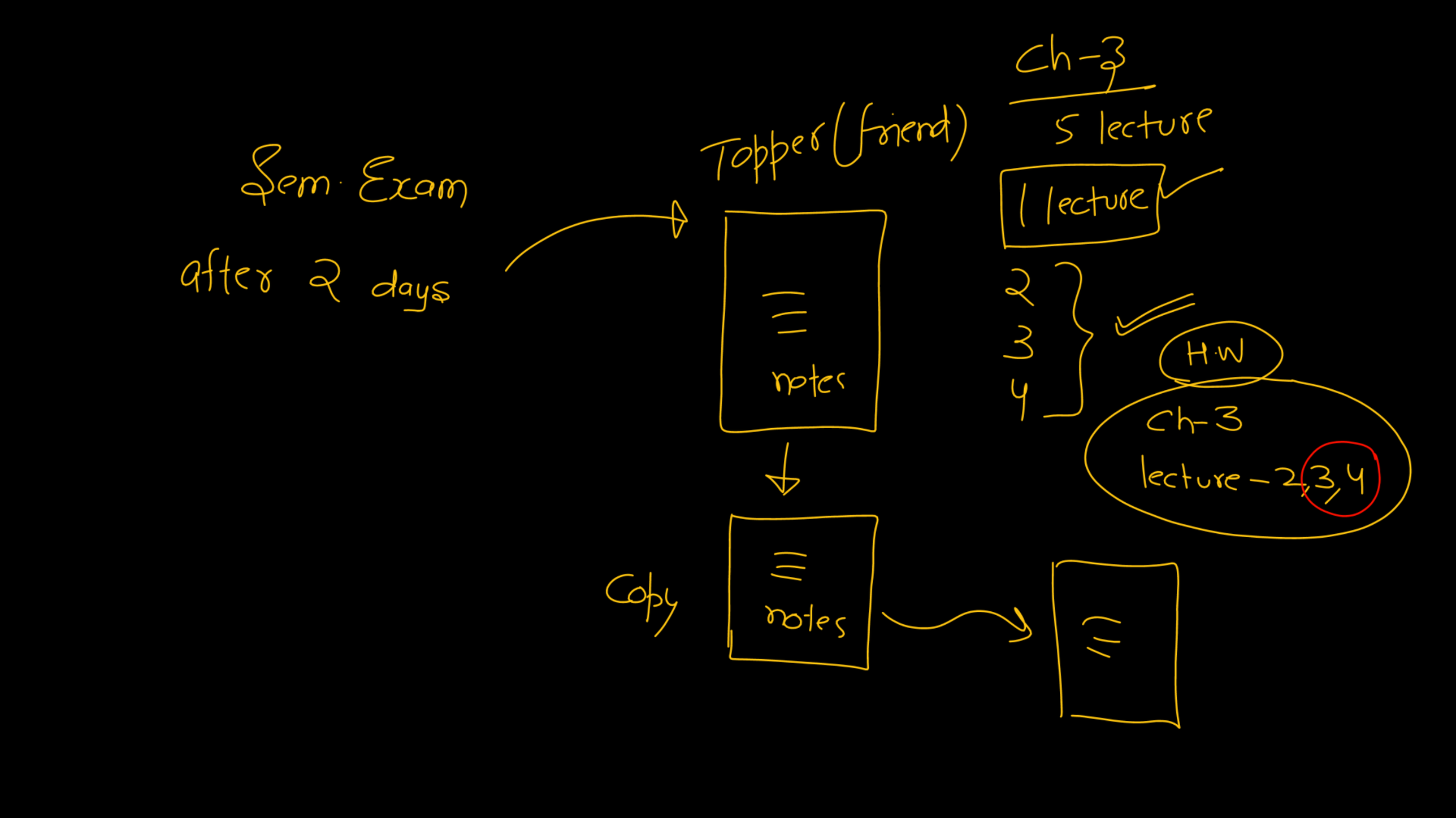
```
int Add(int x, int y)
{
    int sum;
    sum = x+y;
    return sum;
}
```

```
#include<stdio.h>
void swap(int , int);
void main(){
int a=10,b=20;
printf("a=%d,b=%d",a,b);
swap(a,b);
printf("a=%d,b=%d",a,b);
}
```

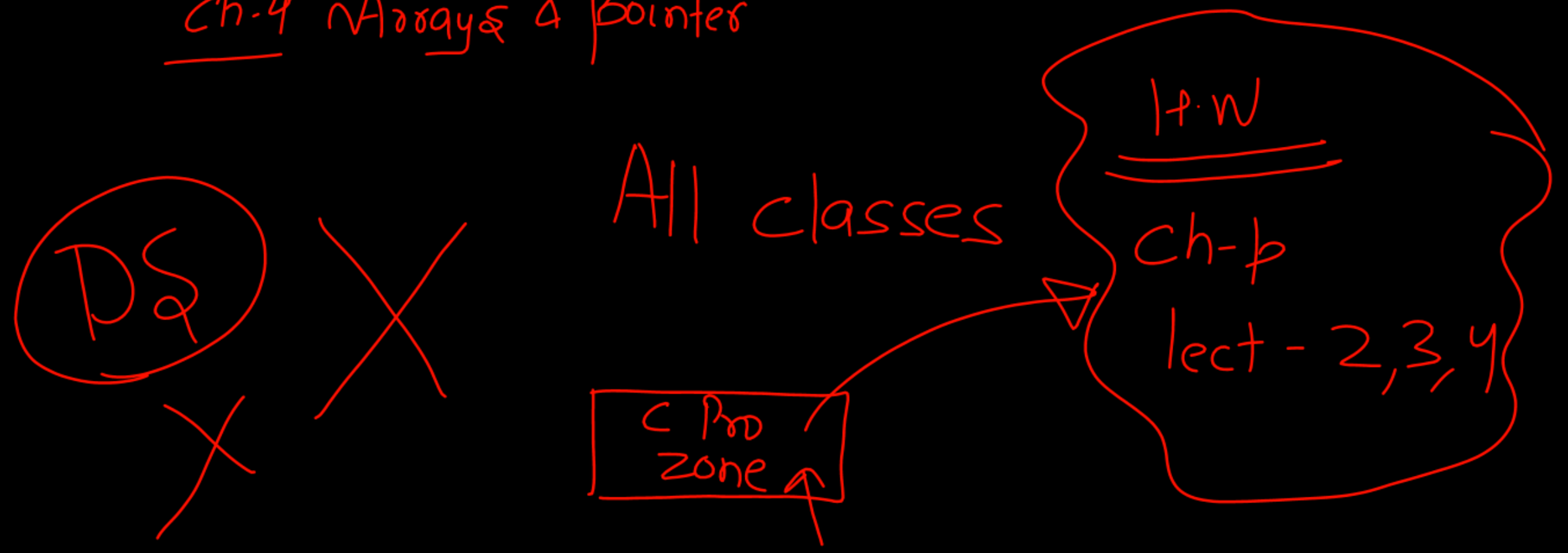
```
void swap(int x, int y)
{
int temp;
temp = x;
x = y;
y = temp;
}
```




```
a=10,b=20 a=10,b=20
```


Sem. Exam
after 2 days
Topper (friend)
notes
Copy
notes
Ch-3
5 lecture
1 lecture
2
3
4
H.W
Ch-3
lecture - 2, 3, 4

ch-4 Arrays & Pointer




THANK YOU GW
SOLDIERS !