CS & IT ENGINEERING

Programming in C

Functions & Storage Classes

Lec- 01

By- Pankaj Sharma sir

TOPICS TO BE COVERED

- Functions-1

Function

$$\left.\begin{array}{l}\text{printf( )}\\\text{scanf( )}\end{array}\right\}\text{Built-in functions}$$

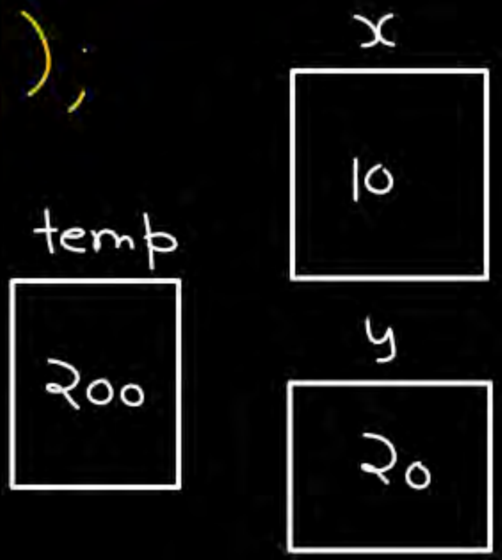
$\boxed{\text{use}}$

re usability

Incomplete code

```
#include<stdio.h>

void main(){

        int a=10, b=20, answer;
                       200              10  20
    answer =  satishsir( a, b );
   printf("%d", answer);

        }
```

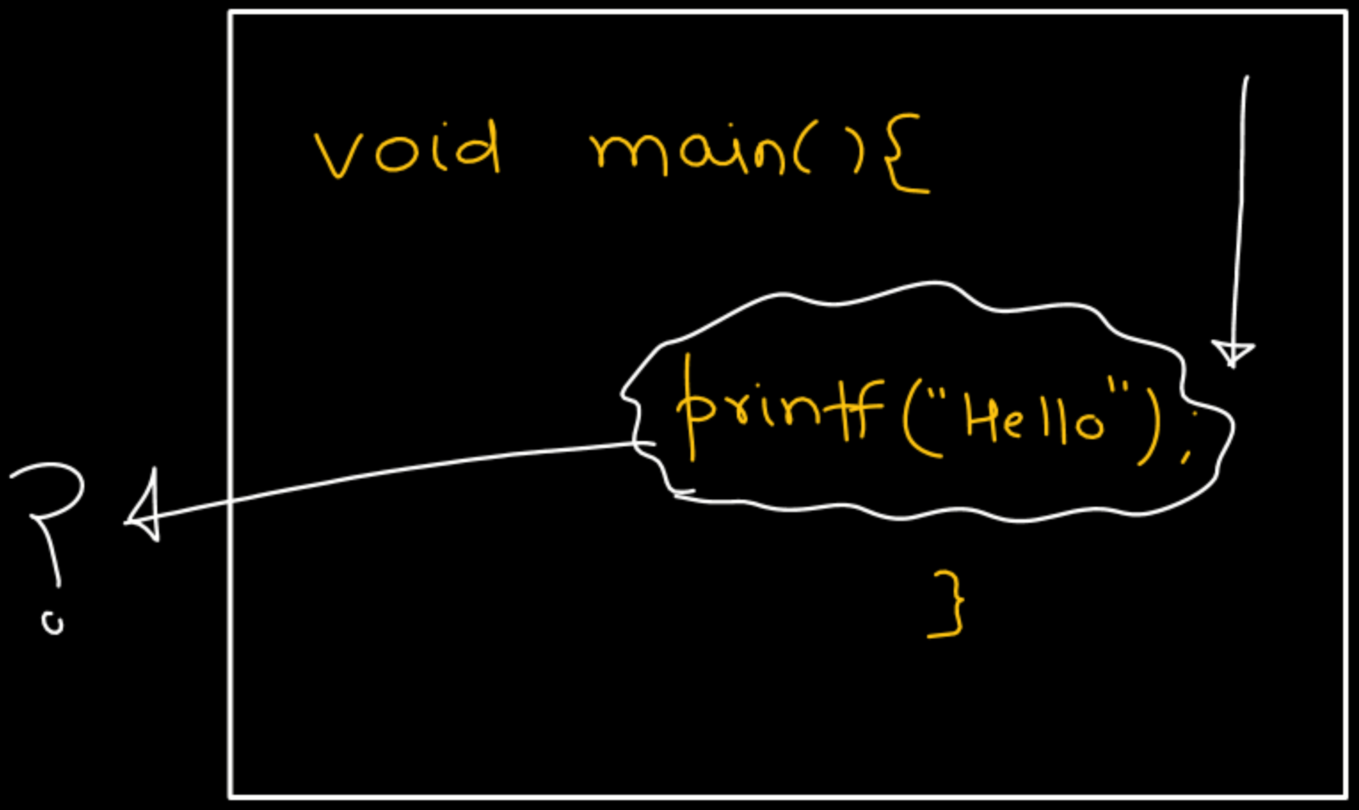
```
satishsir( int x , int y )
  {
   int temp;
   temp = x + y ;
   return temp ;

  }
```



```
#include<stdio.h>

void main(){

        printf("%d",a);

            }
```

→ use without declaration

define



```
void main(){

    printf("Hello");

}
```

(i) Compilation

(ii) Execution

Compilation : Top to bottom

```
#include<stdio.h>
void main(){

      pf("Hello");
      }
```

$\rightarrow$ To avoid C.E

```c
#include<stdio.h>

void main(){

        int a=10, b=20, answer;

    answer = multiply(a,b); //call/use

        printf("%d",answer);

      }
```

define

```c
int multiply(int x, int y)
  {

        int temp;

        temp = x*y;

        return temp;

  }
```

forward declaration (Info → compiler)

(prototype)

```
#include<stdio.h>
int multiply(int,int);
void main(){

      int a=10, b=20, answer;

    answer = multiply(a,b); //call/use

       printf("%d",answer);

     }
```

define

```
int multiply(int x, int y)
{

      int temp;

      temp = x*y;

      return temp;

}
```

```
#include<stdio.h>
int multiply(int x, int y)
{
    int temp;
    temp = x+y;
    return temp;
}
void main(){
    int a=10, b=20, answer;
    answer = multiply(a,b);
    printf("%d", answer);
}
```

$\boxed{\text{int multiply(int x, int y)}} \rightarrow$ function header

int multiply(int, int)

by-default signed

```
short i = 10;
```

```
short int i = 10;
```

by-default
$\Rightarrow$ int

```
#include<stdio.h>

multiply ( int , int );

void main(){

    int a=10, b=20, ans;

    ans = multiply(a,b);

    pf("%d", ans);

    }
```

```
int multiply(int x, int y)
    {
        int temp;

        temp = x*y;

        return temp;

    }
```

by-default ⇒ int return type

College → 1st day दोस्ती

short i = 10;

```
#include<stdio.h>

void main(){

    int a=10, b=20, ans ;

    ans = multiply(a,b);

    pf("%d", ans);

  }
```

save

Happy

The return type of multiply is int

```
int multiply(int x, int y)
{
    int temp;

    temp = x*y;

    return temp;

}
```

```
#include<stdio.h>
void main(){
    int a = 3;
    double b ;
    b = fun(a);
    pf("%f", b);
  }
```

The return type of fun( ) is int.

Mismatch

Error

```
double fun(int x)
{
  double y = 10.2;
  return x*y;
}
```

```
#include<stdio.h>
void main(){
    int a = 3;
    char b;
    b = fun(a);
    pf("%c", b);
  }
```

save

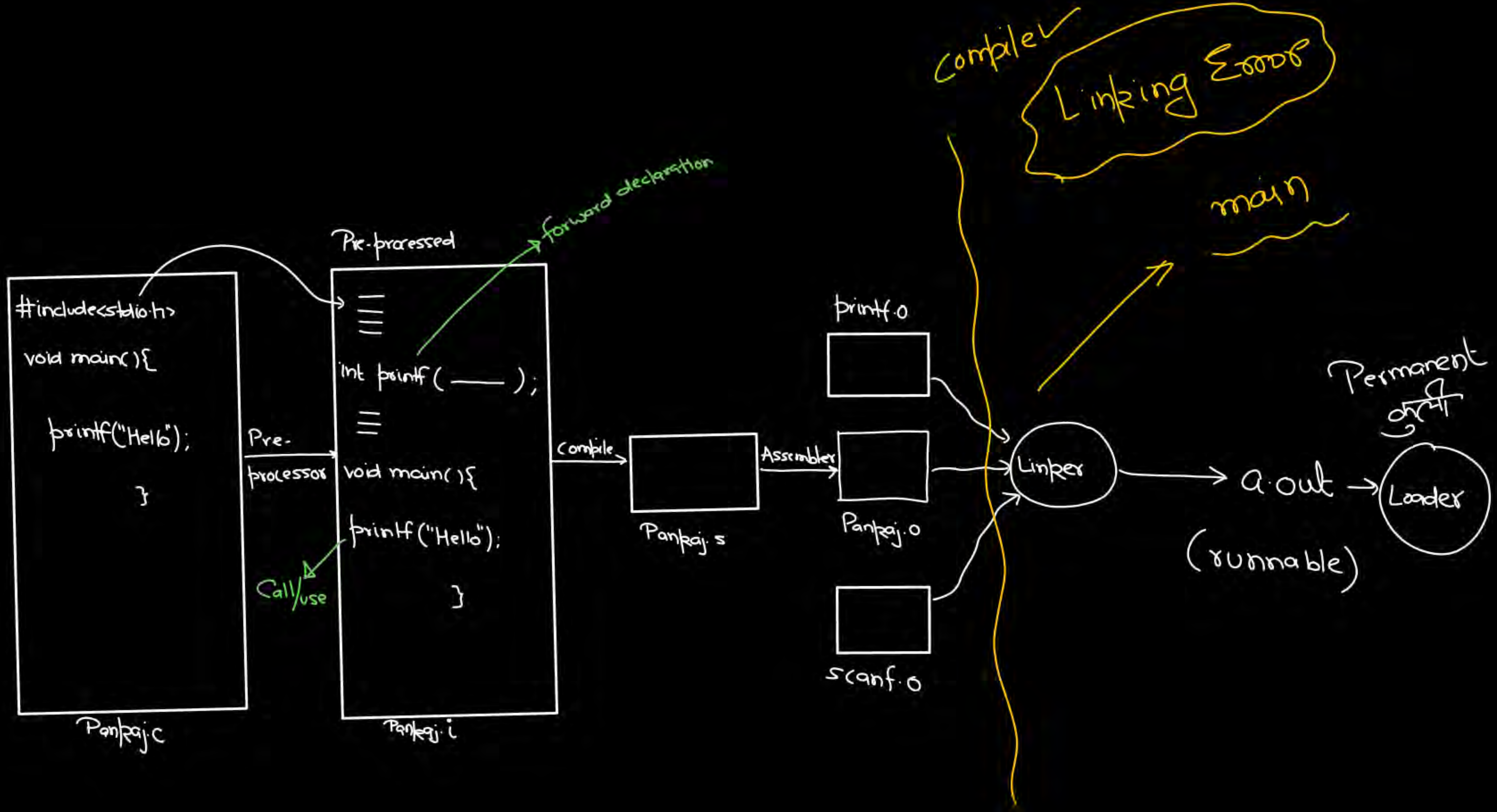
The return type of fun is integer

Mismatch

```
char fun(int x)
  {
   char y = 65;
   return x*y;

}
```

```
#include<stdio.h> forwar declation
 void main(){

    printf("Hello"); //call
       }
```

pf ⇒ define ?


#include<stdio.h>
void main(){
printf("Hello");
}
Pankaj.c
Pre-
processor
Pre-processed
int printf(____);
void main(){
printf("Hello");
}
Pankaj.i
forward declaration
Call/use
compile
Pankaj.s
Assembler
printf.o
Pankaj.o
scanf.o
Compiler
Linker
Linking Error
main
a.out
(runnable)
Loader
Permanent
जमीं

functions

printf( ) ✓

scanf( ) ✓

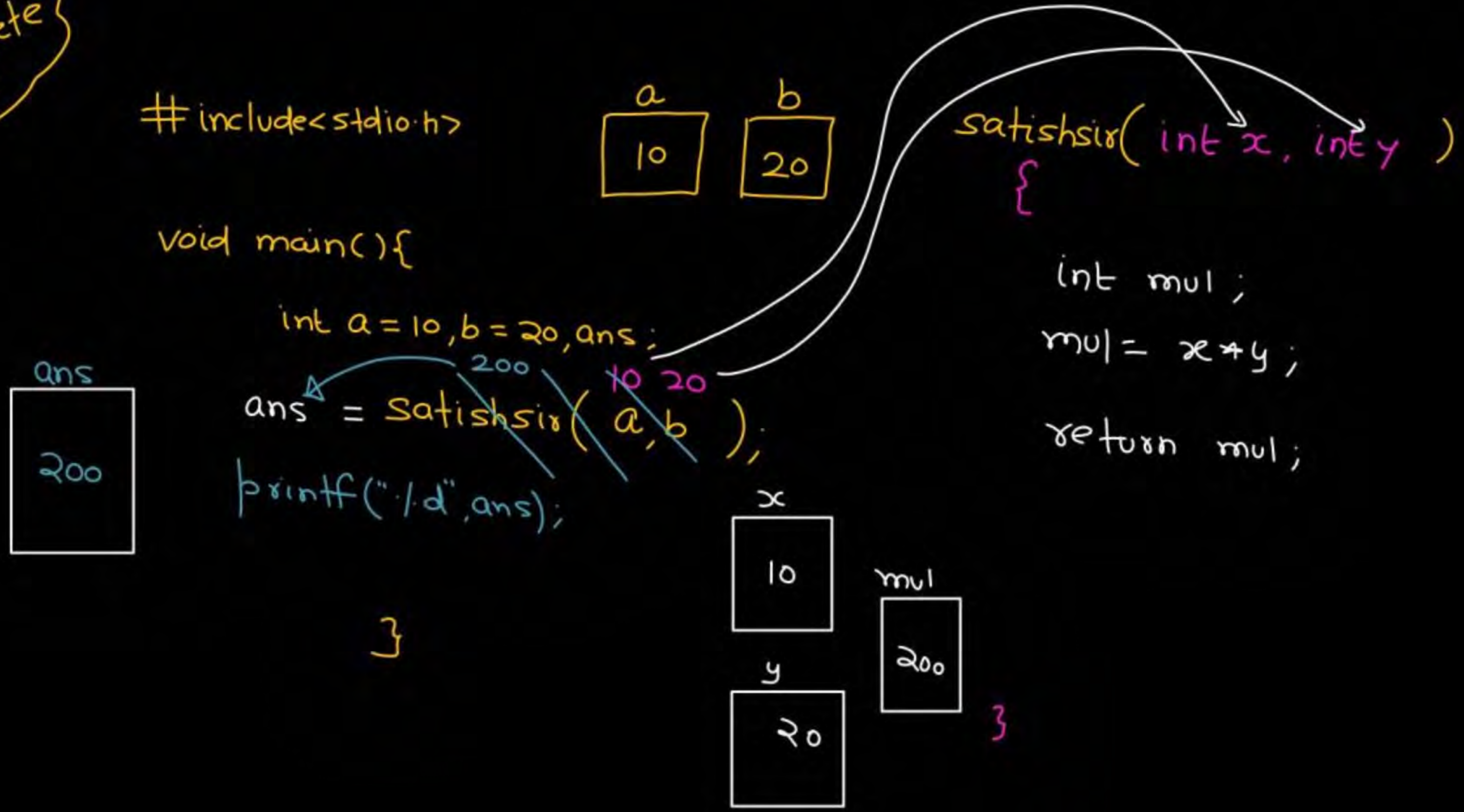
we used them

Code reusability

Incomplete code

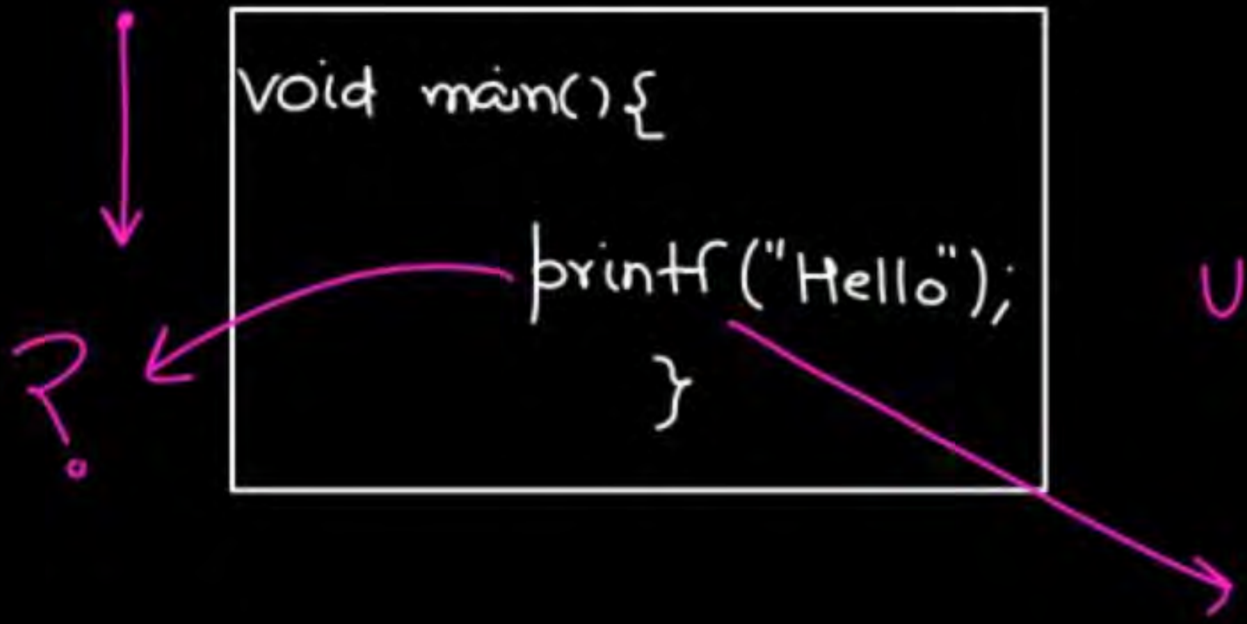
```
#include<stdio.h>

void main(){

    int a=10,b=20,ans;
                        200   10 20
    ans = satishsir( a, b );
    printf("%d",ans);

}
```

```
satishsir( int x, int y )
{

    int mul;
    mul= x*y;
    return mul;
}
```



```
#include<stdio.h>
void main(){

    printf("%d",a);
    }
```

Use a without declaration → compiler info.



```
void main(){
        printf("Hello");
      }
```

using printf

Compilation

Executation

related info $\Rightarrow$ header file

```c
#include<stdio.h>

void main(){

    int a=10,b=20,ans;

    ans = Multiply(a,b);

    printf("%d",ans);

    }
```

→ use/call

?

To avoid any C.E

⇒ forward declaration

definition/body of func.

```c
int Multiply(int x, int y)
{

    int res;

    res = x*y;

    return res;

}
```

```
#include<stdio.h>

int Multiply(int,int);    //forward declaration
                               prototype

void main(){

    int a=10,b=20,ans;

    ans = Multiply(a,b); //call

    printf("%d",ans);

          }
```

define/body

```
int Multiply(int x,int y)
  {
   int res;
   res = x*y;
   return res;
  }
```

```
# include<stdio.h>

  define/body
int Multiply(int x, int y)      function header
  {
    int res;
    res = x*y;
    return res;
  }
void main(){

     int a=10,b=20,ans;

     ans = Multiply(a,b); //call

     printf("%d",ans);

          }
```

by default

```
short i = 10;

short int i = 10;

signed short int i = 10;

signed short i = 10;
```

All are same

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

        }
int mul(int x, int y){

            return x*y;

        }
```

the return type of
mul function is
int

Happy

```c
#include<stdio.h>

void main(){
        int x;
        x = fun(10);
        printf("%d",x);
    }
double fun( int y){
    double temp=12.0;
    return temp*y;
    }
```

info save

$\boxed{\text{return type of fun function is int}}$

double → Mismatch

Error

implicit

Compilation

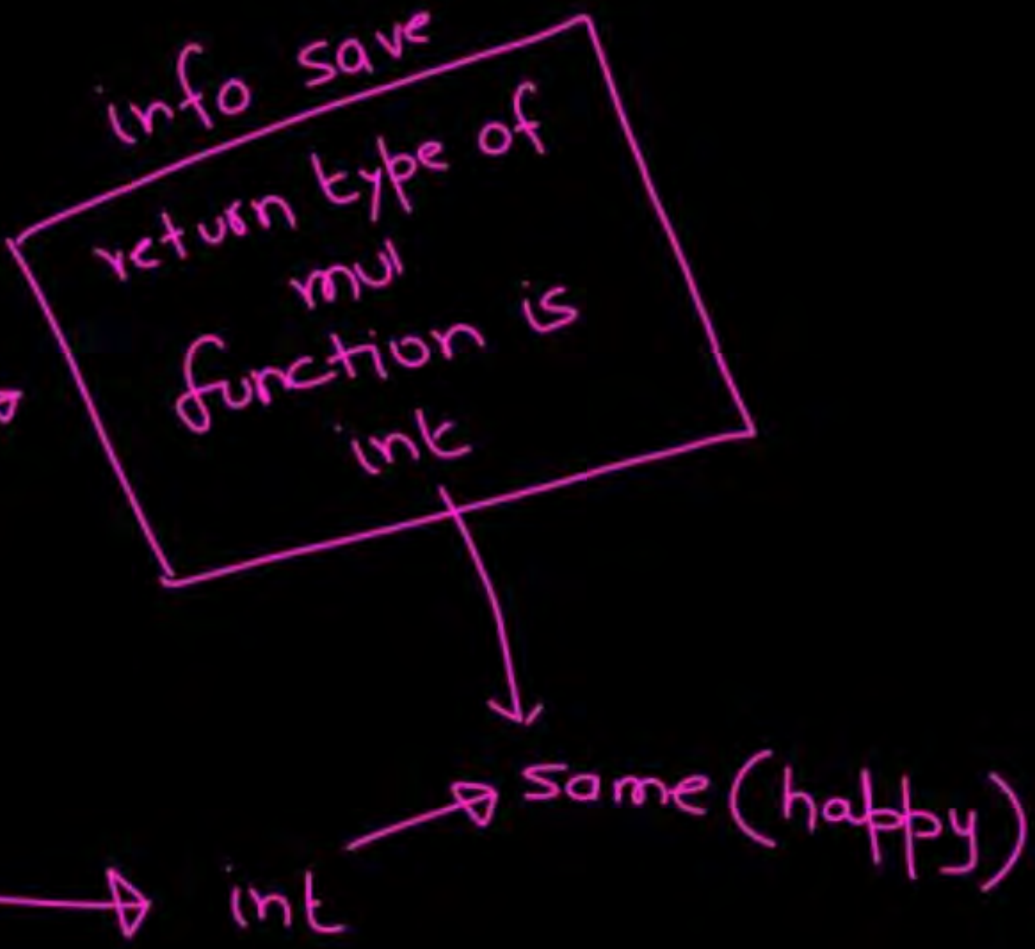
Top to bottom ↓

main से कोई लेना देना नहीं है

Execution

↳ main ✓

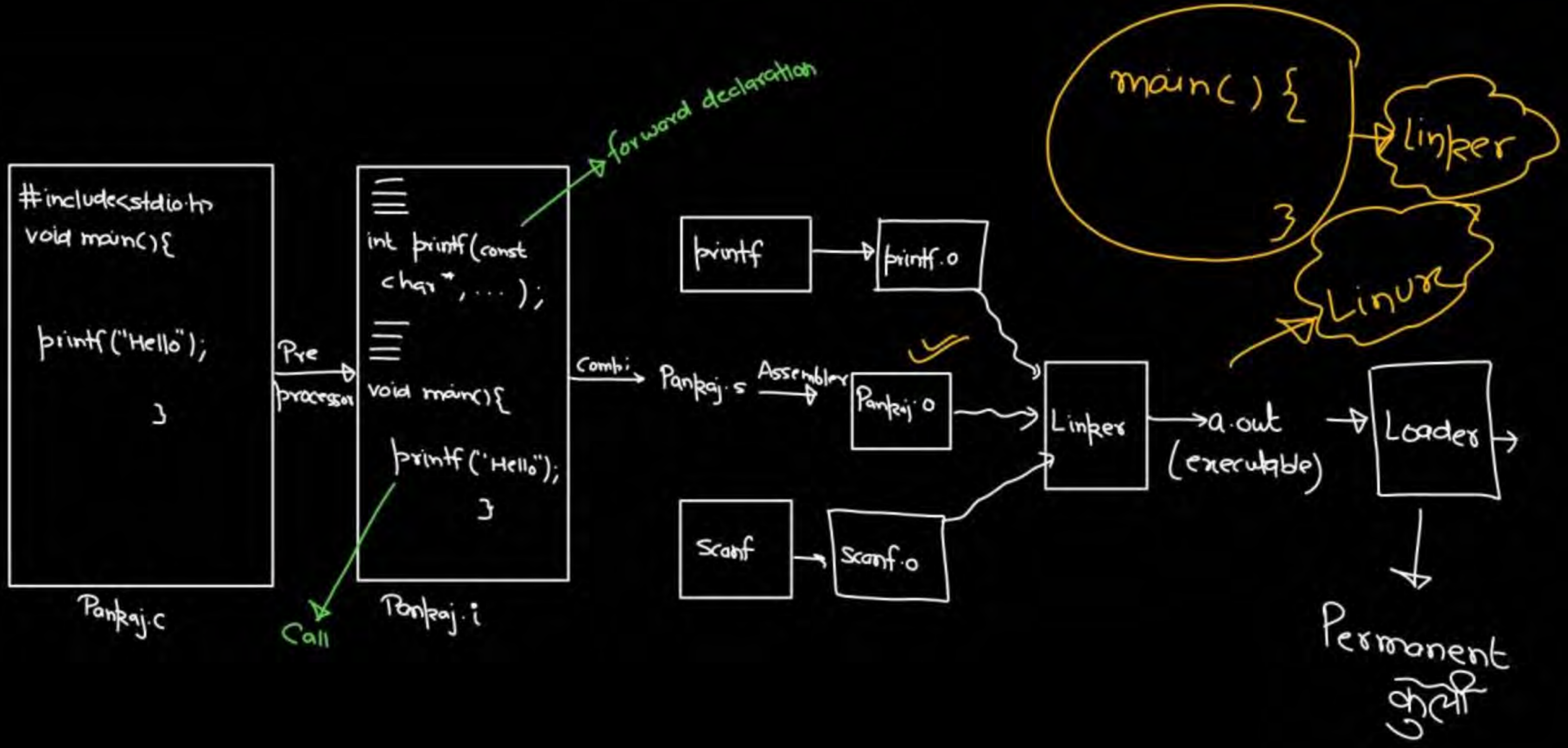
```c
#include<stdio.h>
void main(){

        int a=10,b=20,ans;

        ans = mul(a,b);

        printf("%d",ans);

        }
int mul(int x, int y){

        return x*y;

        }
```



forward declaration

```
#include<stdio.h>
void main(){

    printf("Hello"); //call

}
```

Compile ✓



```
#include<stdio.h>
void main(){

 printf("Hello");

        }
```

```
int printf(const
 char*, ... );

void main(){
  printf("Hello");
        }
```

```c
#include<stdio.h>

int mul(int x, int y)
  {

     return x*y;

  }
```

Compile ✓

Execute ✗

Linking Error

```
#include<stdio.h>
void main(){

        printf("Pankaj");

    }
```

use ✗

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

use ✓

pf $\rightarrow$ return a value

```
#include<stdio.h>
void main(){

    int a=10,b=20;

    mul(a,b);

          }
```

```
int mul(int a, int b)
   {
     int temp;
     temp=a*b;


   }
```

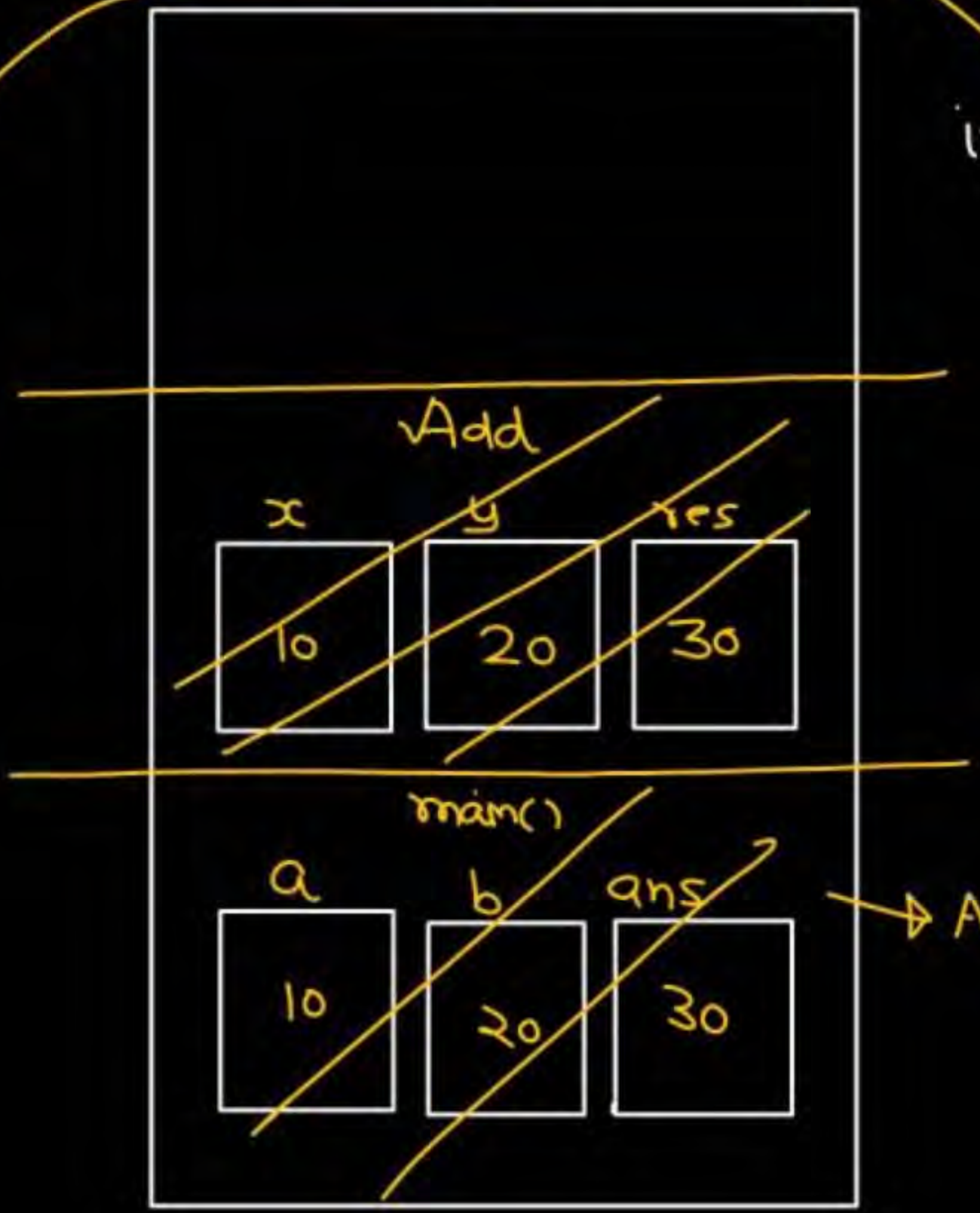
Not returning anything

```c
#include<stdio.h>
void mul(int, int);

void main(){
        int a=10, b=20 ;
        mul(a, b);
        }
```

```c
void mul(int x, int y)
{
    int temp;
    temp = x+y;
    printf("%d", temp);
}
```

How function works

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



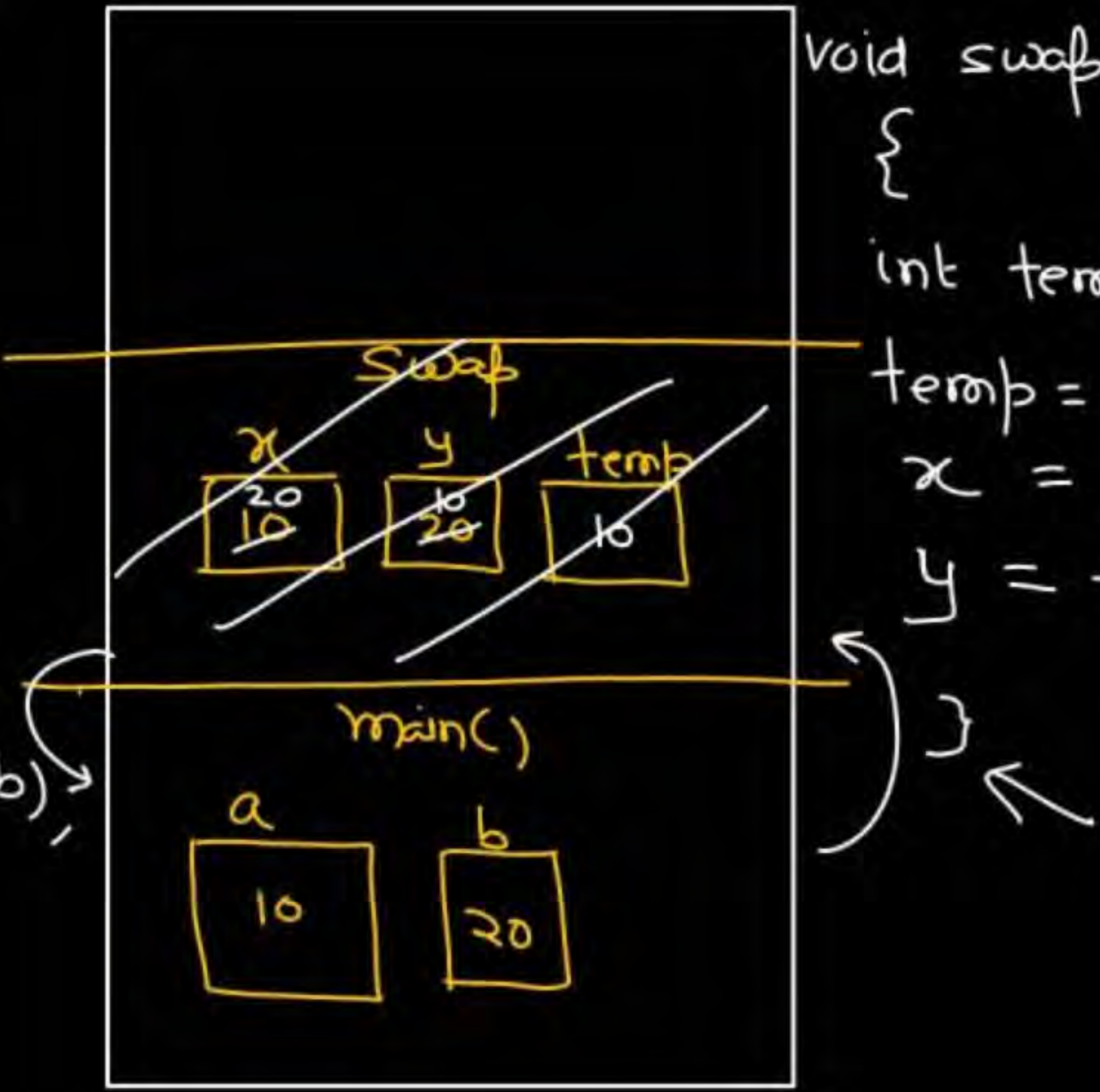
```
int Add(int x, int y)
{
   int res;
   res = x+y;
   return res;
}
```

COPY

```
#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 and b=%d",a,b);
}
```

10 20 ✓

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



```
#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 and b= %d",a,b)
}
```

⇒ Calling

10 20 ✓

$\overset{10}{a}$, $\overset{20}{b}$ ⇒ actual values (Actual Parameters)

Called

formal parameters

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

main ⟶ swap

THANK YOU GW
SOLDIERS !