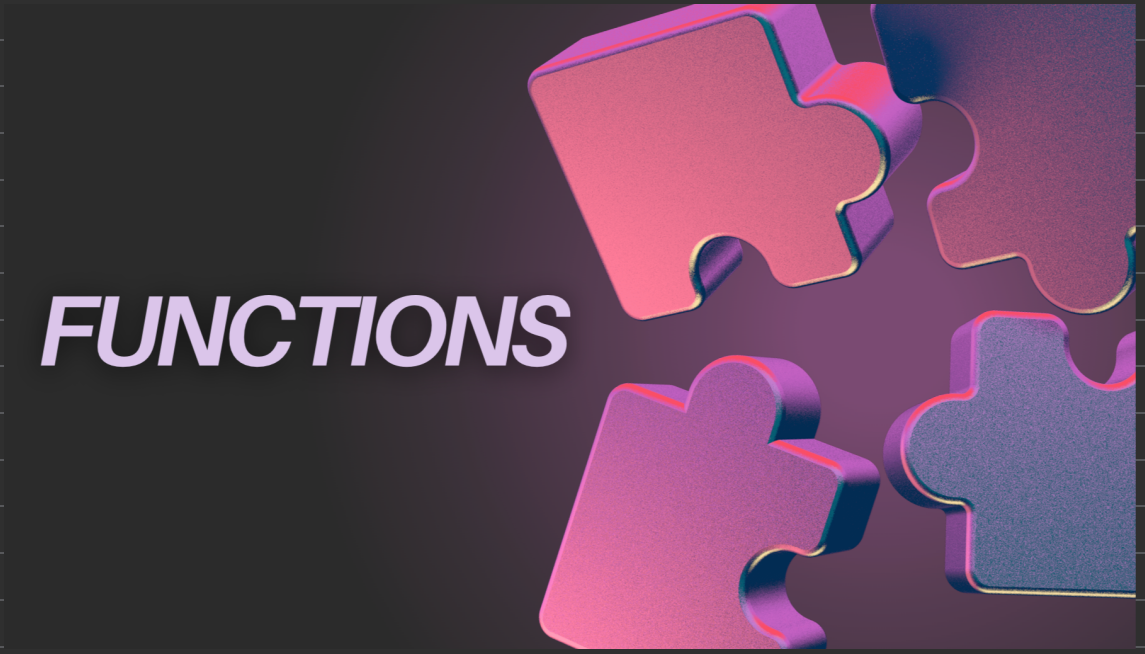
"खुल जाएंगे सभी रास्ते, तू रुकावटों से लड़ तो सही,
सब होगा हासिल, तू अपनी ज़िद पर अड़ तो सही।"



Made with Goodnotes

main() — compulsory function

- program = set of functions
- function = set of instruct^

Made with Goodnotes

Why we need Functions

Two No. every time & we have to swap it

duplicate code

```
main()
{
    x, y
    temp = x
    x = y
    y = temp
    pf(x, y)

    a, b
    temp = a
    a = b
    b = temp
    pf(a, b)

    m, n
    temp = m
    m = n
    n = temp
    pf(m, n)
}
```

Made with Goodnotes

```
swap ( x, y)
{   temp = x
    x = y
    y = temp
}
```

```
main()
{
    x, y
    swap (x, y)

    a, b
    swap (a, b)

    m, n
    swap (m, n)
```

Made with Goodnotes

What is Functions

It is a set of instructions, which mainly performs certain task.

Function Declaration & Defination

Made with Goodnotes

swap

add

find total student

findTotalStudent()

(camel case)

Made with Goodnotes

* Declaration

Syntax :

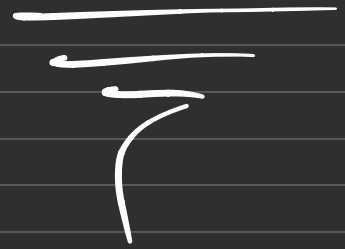
```
Return Type    nameOfFunction ( )
{

}
```

Made with Goodnotes

* Defination

swap ( )

{



}

Made with Goodnotes

Return type

Name of functn

What is Functions call

```
void add()
{
    int x=10, y=20
    ans = x+y
    pf(ans)
}

main()
{
    int a=10
    a = a+2
    add() // calling
    pf(a)
```

30 12

Activation Stack Record

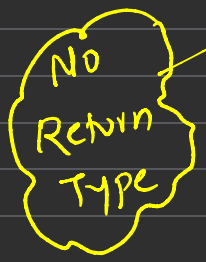
Made with Goodnotes

Types of Functions

- No return Type & No Argument
- No return Type & With Argument
- With return Type & No Argument
- With return Type & With Argument

Made with Goodnotes

No return Type & No Argument



void NameOfFunction( )
{
}

```
void add()
{
    int x=10, y=20
    ans = x+y
    print(ans)
}

main()
{
    add()
}
```

Made with Goodnotes

No return Type & With Argument

Syntax:

```
void   functionName ( arg1 , arg2 , arg3 ..... )
{


}
```

Made with Goodnotes

Argument / Parameters

```
void add ( int a , int b )
{
    int ans = a + b
    printf ( ans )
}

main()
{
    int x , y
    cin >> x >> y
    add ( x , y ) //
}
```

x = 90
y = 70

void add (int a, int b)

add (90, 70)

Made with Goodnotes

With return Type & No Argument

int / float / String / Array / bool / char / float

Syntax :

```
Returntype  functionName ( )
{

    return val
}
```

Made with Goodnotes

Return Type

No Argument

```
int add ()
{
    int x = 90, y = 60
    return (x+y)
}

main()
{
    int a = 10
    ans = add()
    a = a + ans
    pf(a)
}
```

150

160

a

10

NOTE:

Value returned by function, is returned where function is called

Made with Goodnotes

With return Type & With Argument

Syntax :

```
Return_type  functionNm (arg1, arg2, arg3.....)
{

    return (val)
}
```

Made with Goodnotes

```
bool isVowel ( char ch)
{
    if ( ch == 'a' || ch == 'e' || ch == 'i' , ch == 'o'
        || ch == 'u')
        return true

    else
        return false

}
```

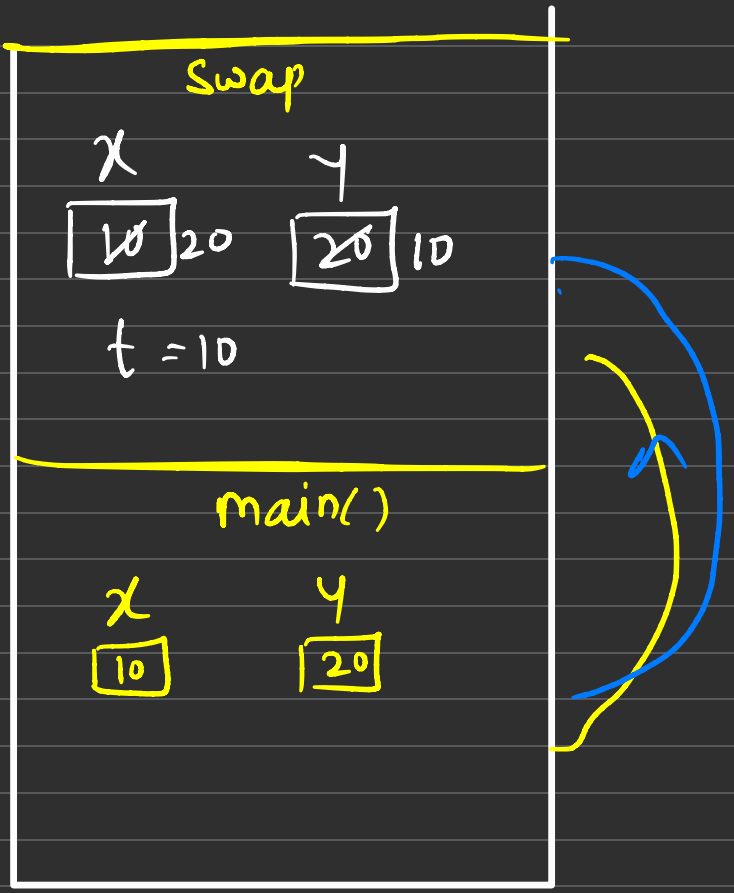
```
main()
{
    ans = isVowel(ch)
    if (ans)
    {
```

Made with Goodnotes

Call by Value Vs Call by Reference (VV imp)

Call by Value.

```
swap ( int x, int y)
3      t = x
       x = y
4      y = t

main()
?   x = 10, y = 20
    swap (x,y)
{
```



Made with Goodnotes

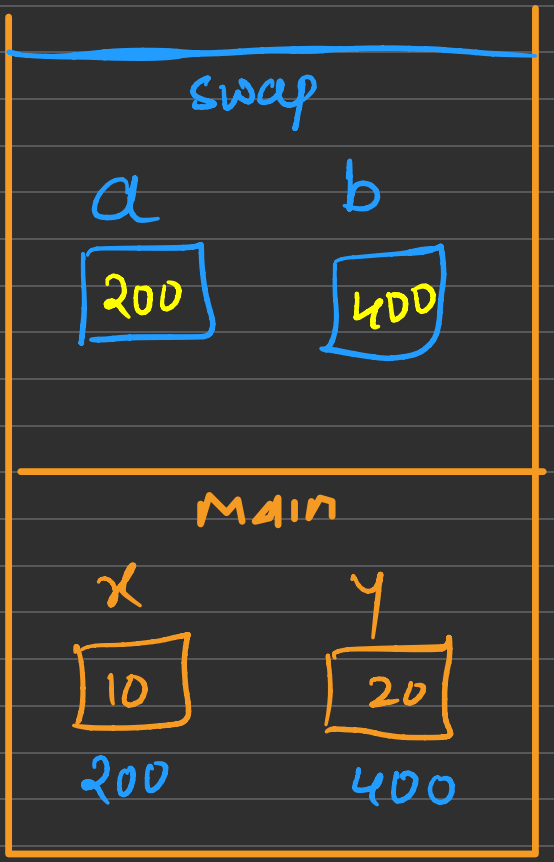
```cpp
// function definition
void swap(int x, int y){

    int temp = x;
    x = y;
    y = temp;
}
```

```cpp
int main(){

    // two number to
    // no return with

    int x, y;
    cin >> x >> y;

    cout << "Before Swaping\n";
    cout << x << " " << y;

    swap(x, y);

    cout << "\nAfter Swaping\n";
    cout << x << " " << y;

    cout << "\n";
    return 0;
}
```

Made with Goodnotes

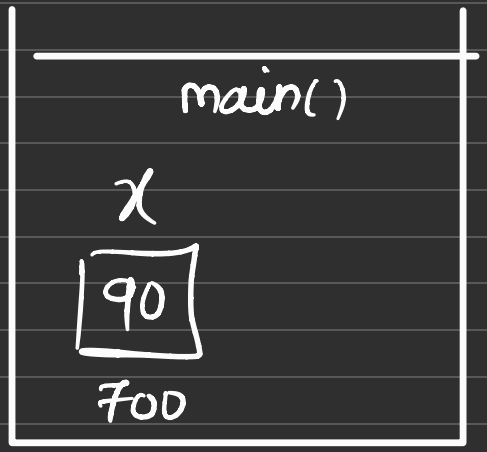
Call of Refrence ⟶ Address

```
void swap ( int &a, int &b)
{
    t = a
    a = b
    b = t
}

main()
{
    x = 10, y = 20
    swap (x, y)
}
```



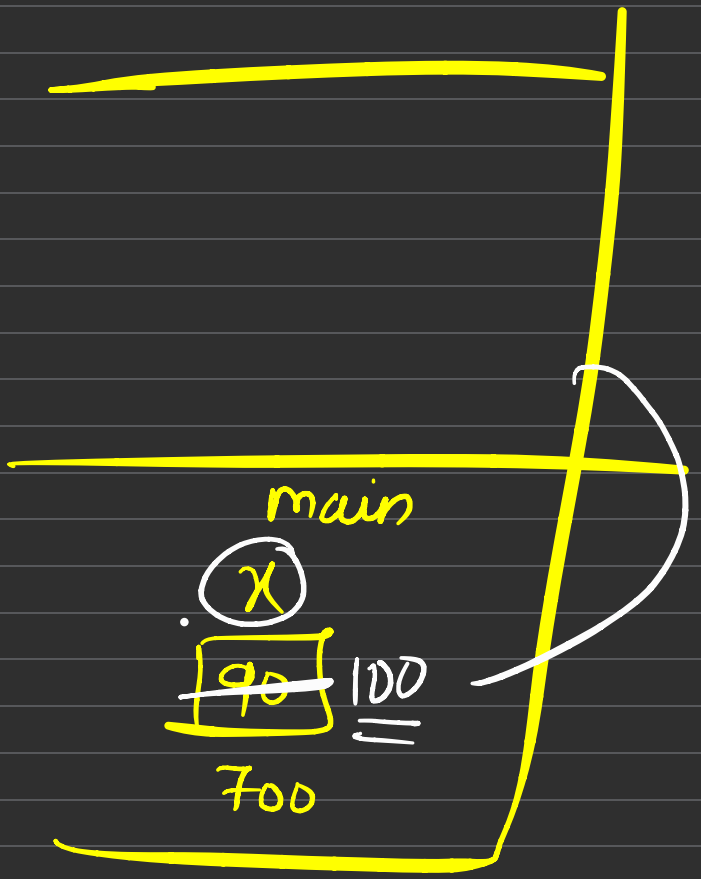
Made with Goodnotes

```
void fun ( int x)
{
    x = x+10
}

main()
{
    x = 90
    fun(90)
}
```



Made with Goodnotes

```
void fun ( int &x)
{
    x = x + 10
}

main()
{
    x = 90
    fun(x)
}
```



Made with Goodnotes