unacademy

Default Arguments

Comprehensive Course On C++

Pankaj Sharma • Lesson 4 • May 15, 2024

unacademy

CS & IT Engineering

C++

Default Arguments

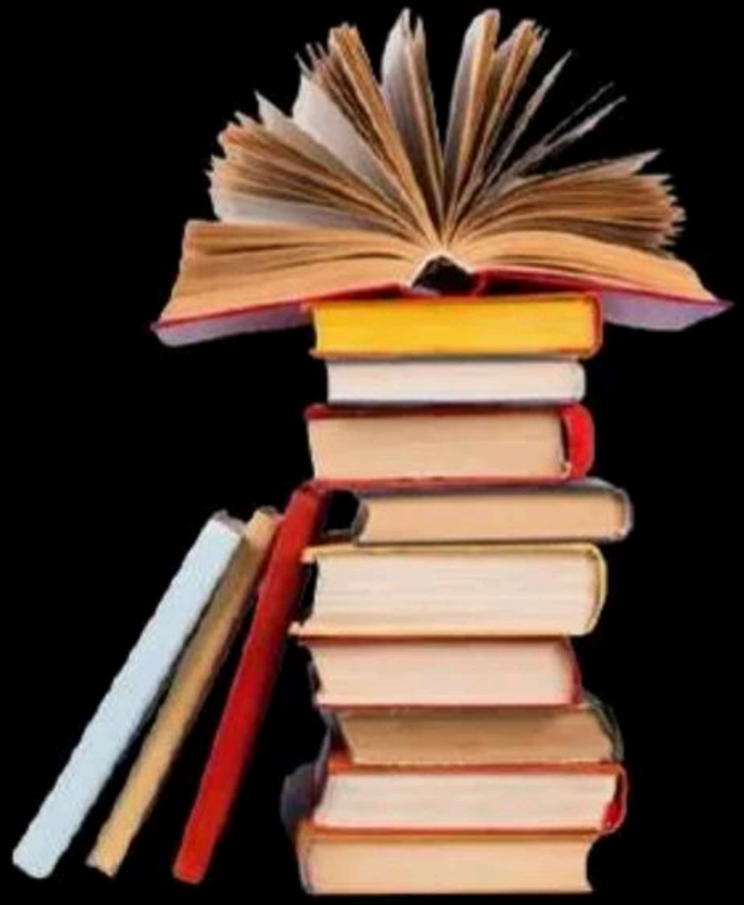
Lecture 04

By- Pankaj Sir

unacademy

Topics

to be covered



1. Default arguments

unacademy

⇒ Refrence Variable

↳

C → i) Formal arg. (ii) Actual arg.

unacademy

Call by value

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

(10 20 — actual arg)

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

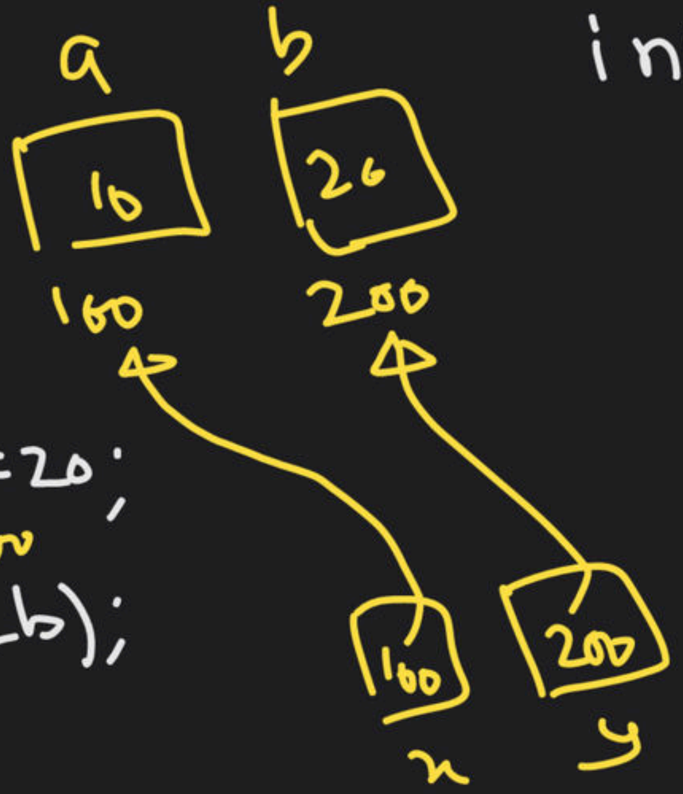
(formal arg)

simple variable / value type

unacademy

Call by address / __

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




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

pointer variable

In C lang.

1. Simple variable
2. pointer variable

C++ is a superset of C lang

C++

1. Simple var.
2. pointer var.
3. Refrence variable.

unacademy

Refrence Variable

```
int x;
int *p;
```

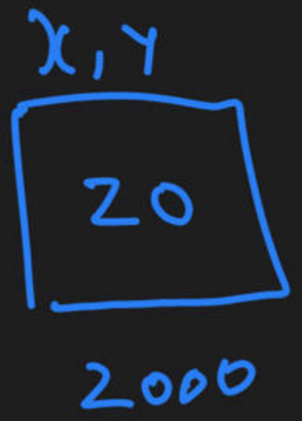
```
int &y
```

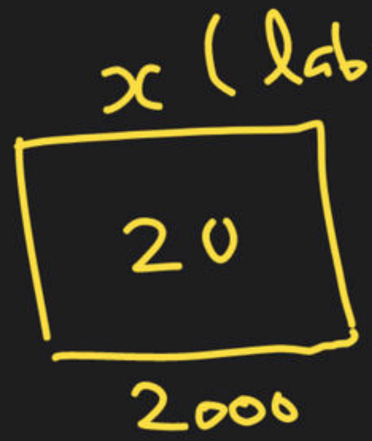
(Not complete)

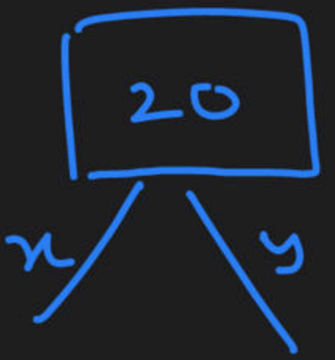
y is a refrence variable

unacademy

```
int x = 20;
```

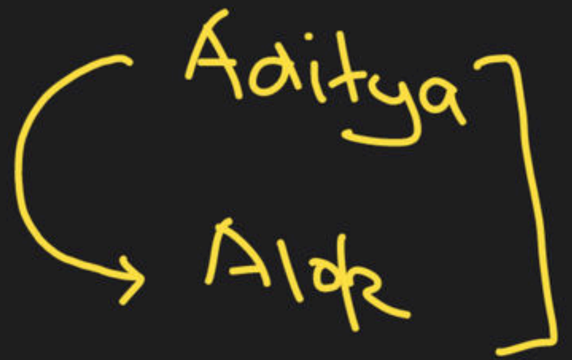







```
int &y = x;
```

y is a reference variable (alias)



⇒ y is another label (name) of x.

unacademy

int &y; → Vd fe laat Pdegi

↳ refrence variable

must be initialize

(with some existing variable)

unacademy

```
int x = 10;
int &y = x;
```

Physical storage X

x, y ⇒ same

1. Can not change.
2. It must be initialized with some existing var/object.
3. Can not be NULL.

Call by value

```
int f(int,int);

Void main(){
    int a=10,b=20,-;
    =
    f(a,b);
    =
    }

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

Call by address

```
int f(int*,int*);

Void main(){
    =
    f(&a,&b);
    =
    }

int f(int *x, int *y){
    =
    int *x=&a;
    =
    }
```

Call by refrence

```
int &y=b;
int &x=a;

Void main(){ int &y=b
    int a=10,b=20;
    f(a,b);
    }

int f(int &x, int &y)
    {
    =
    }
```

Call by value

```
int f(int,int);

Void main(){
    int a=10,b=20,-;
    =
    f(a,b);
    =
}

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

Call by address

```
int f(int*,int*);

Void main(){
    =
    f(&a,&b);
    =
}

int f(int *x, int *y){
    =
}
```

Call by reference

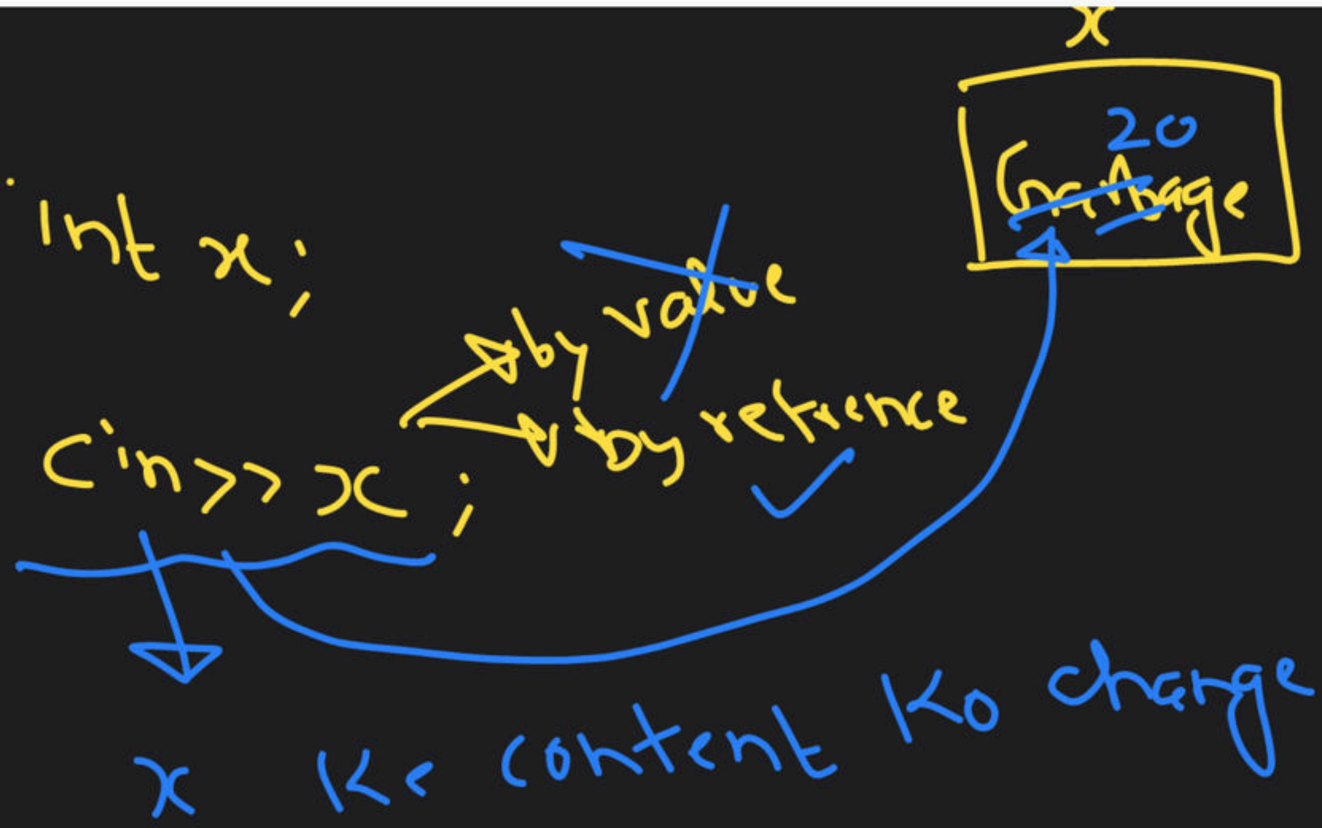
```
int f(int &,int &);

Void main(){
    int a=10,b=20;
    f(a,b);
}

int f(int &x, int &y)
{
    =
}
```

function → change → call by add
                  → call by ref
                  → No
                  → call by value

```
int a=10,b=20;
function bf("%d %d",a,b);
```

call by value
bf ki
// call

X linked list, trees



unacademy

```
int a = 1, b = 2;
int &c = a, d = b;
```

a, b, d → int

c → ref. variable.

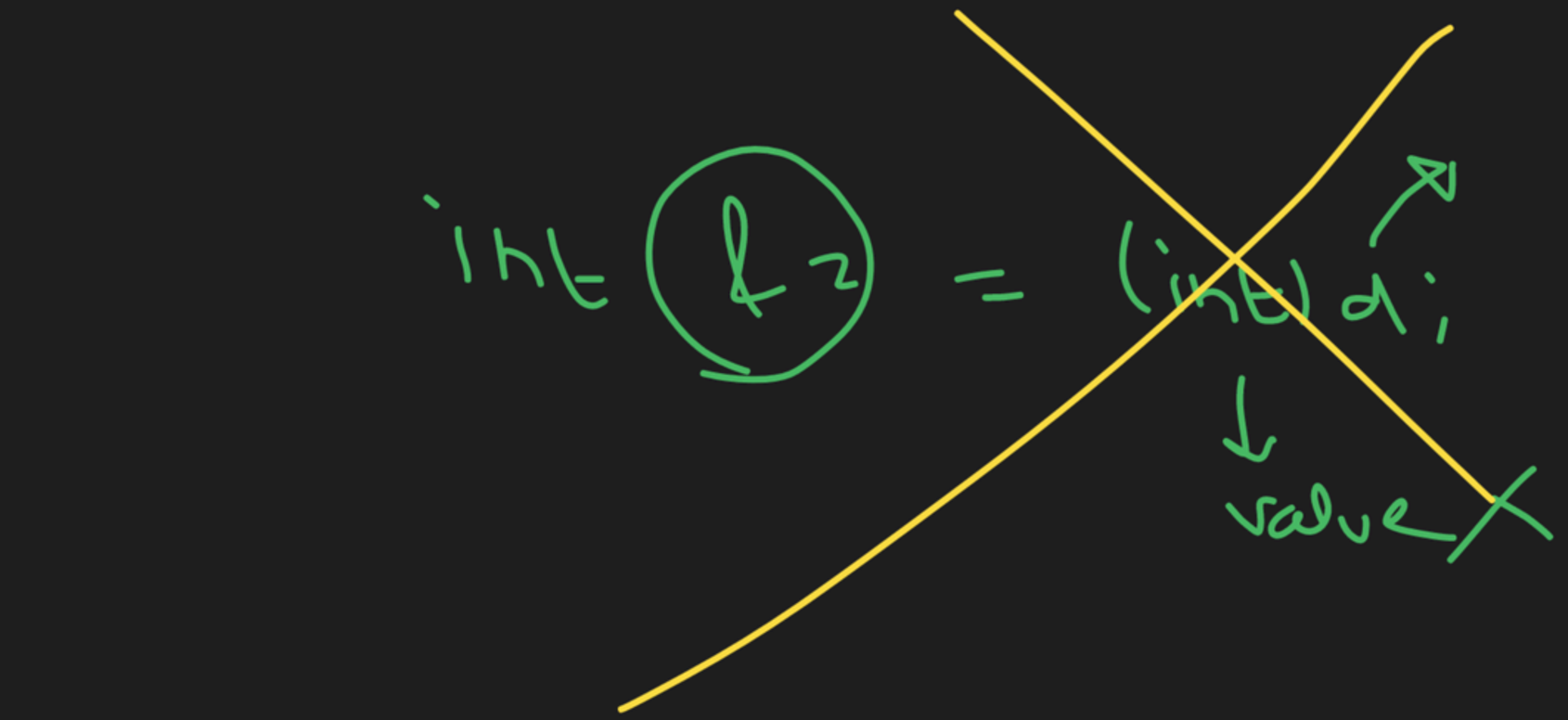
initializer must be an object of same type

① `int i = 10, j = 20;` ✓

② `int &x = 20;` Error

③ `int &y = i + j;` Error

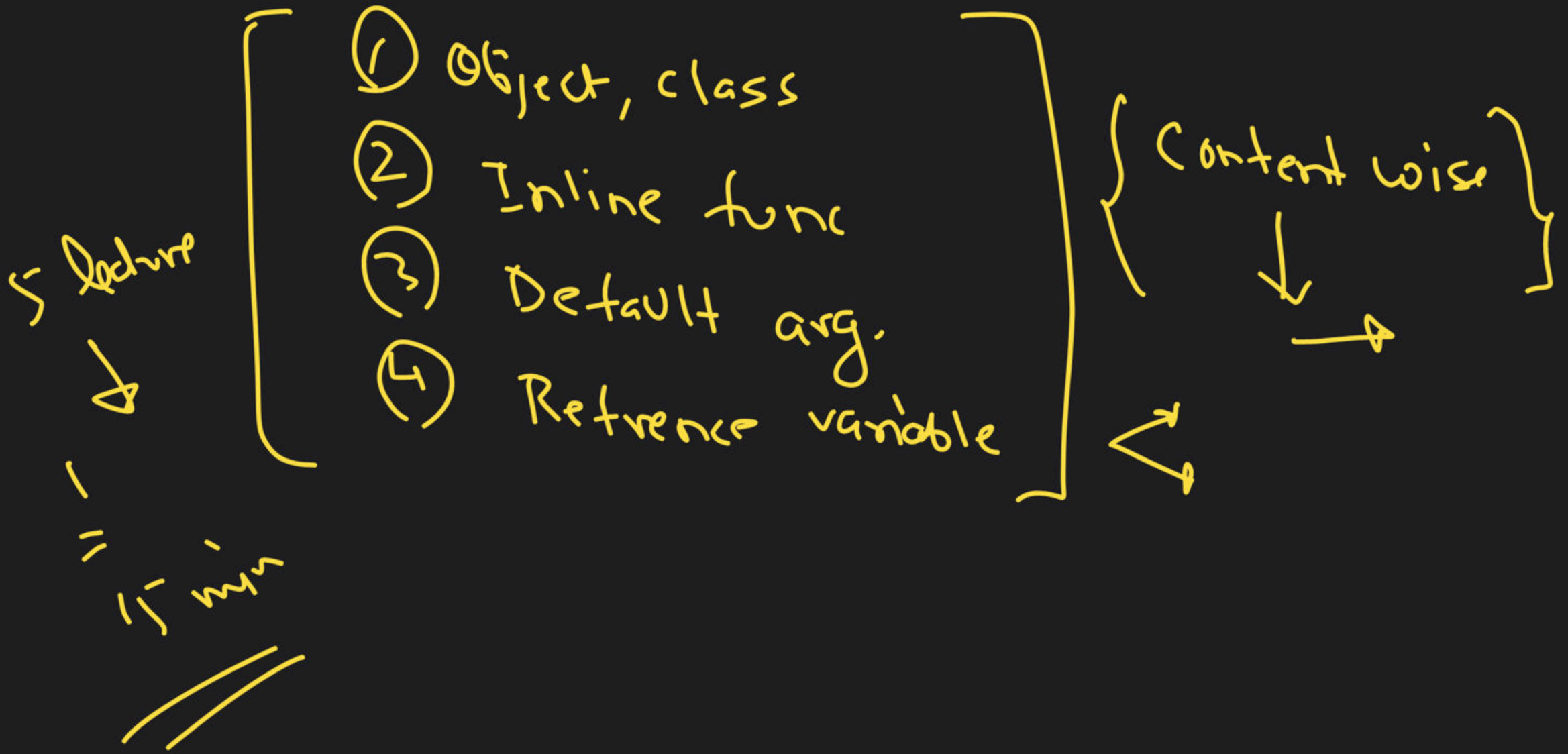
```
double d = 9.8;
int &z = d;
```



unacademy

Refrence variable

1. It must be initialized with already existing var/object.
2. Initializer must be an object/var. of same type.
3. It can not be changed (No divorce, No break-up)
4. It can not be NULL.


5 lecture
1
2
3
4
1. Object, class
2. Inline func
3. Default arg.
4. Refrence variable
Content wise

```
int *x = &a;
int *y = &b;
```

```
int *x = &a;
int *y = &b;
```

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

```
f(int *x, int *y)
{
}
```

unacademy

THANK YOU!

Here's to a cracking journey ahead!