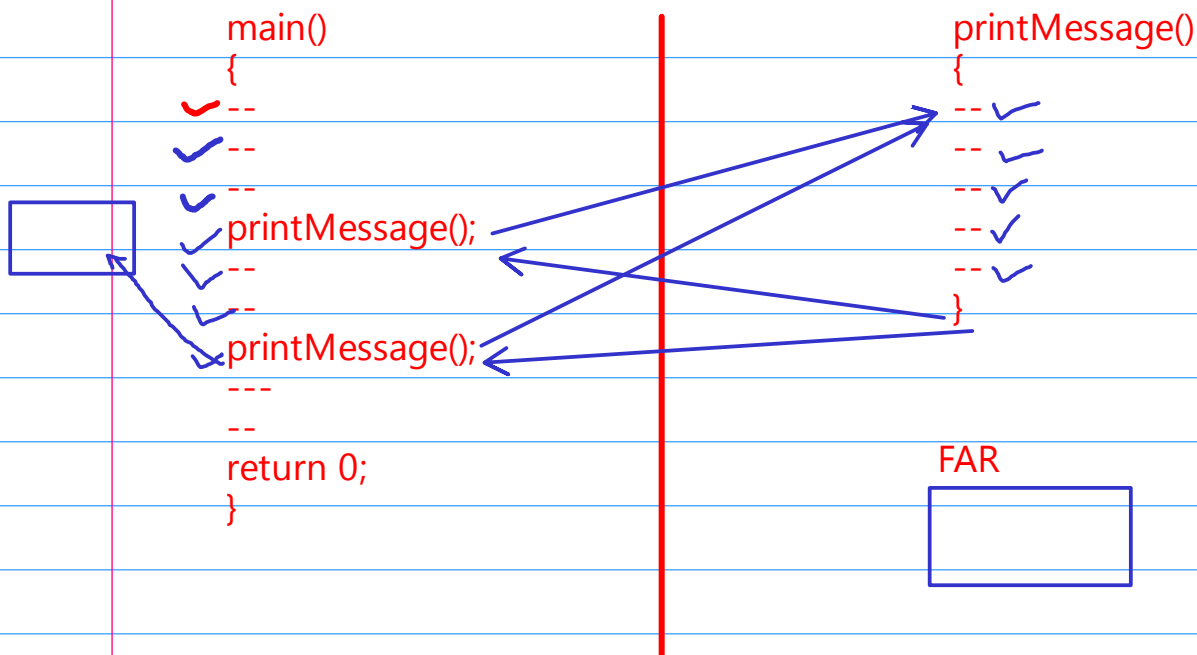


DAC cpp = 100%  
 preCAT CPP = 35%  
 DESD cpp = 70%

CPP= c+oop



- ✓ void printValue(int a) => printValue@int
- ✓ void printValue(int a,int b) => printValue@int,int
- ✓ void printValue(char a) => printValue@char
- ✓ void printValue(int a,char b)=> printValue@int,char ✓
- ✓ void printValue(char a,int b)=> printValue@char,int ✓

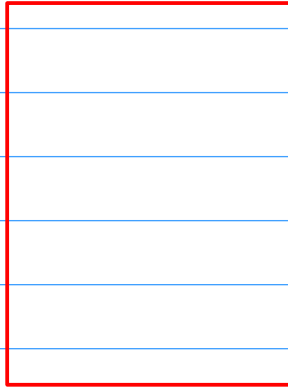
1 / 0 int => 4 => 32 1  
 bool => 1 => 8

char => 1 => 8 => 256  
 wchar\_t => 16 => 2 => 65536

book ✓  
 pages ✓  
 author ✓  
 price ✓  
 pub ✓  
 sal  
 roll\_no ✗

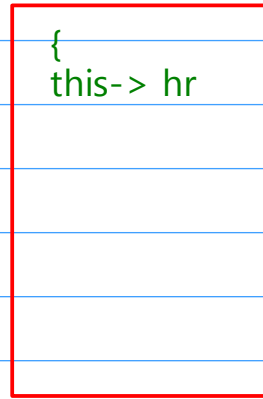
time  
 hr  
 min  
 sec

print()



accept() this=4000

{  
 this-> hr




---

t1	hr	min	sec	t2	hr	min	sec	t3	hr	min	sec
	11	22	33		10	3	20		8	5	33
	2000				3000				4000		

t1.accept()

t2.accept()

t3.accept()

c++ => this  
 java=> this  
 c# => this  
 python => self

## Structure in C

```

struct time {
    int hr, min, sec;
}; //
void accept( struct time *p) {
    scanf("%d:%d:%d", &p->hr,
    &p->min, &p->sec);
}
Main()
{
    struct time t;
    accept(&t);
}
    
```

Diagram for C Structure:

hr	min	sec
11		

Annotations:   
 - **7000** is written next to the `*p` parameter in the `accept` function.   
 - **7000->hr** is written next to the `&p->hr` argument in the `scanf` call.   
 - **7000** is written next to the `&t` argument in the `accept(&t);` call.   
 - A red arrow points from the `struct time t;` line to the memory box.

## class in C++

```

class time {
    int hr, min, sec;
    void accept() {
        scanf("%d:%d:%d", &hr, &min,
        &sec);
    }
}; //end of class
Main()
{
    time t;
    t.accept();
}
    
```

Diagram for C++ Class:

hr	min	sec
12		

Annotations:   
 - **const** is written next to the `time` class name.   
 - **time \* this** is written next to the `&hr` argument in the `scanf` call.   
 - **&this->hr** is written next to the `&hr` argument in the `scanf` call.   
 - **mf** is written next to the `t.accept();` call.   
 - A red arrow points from the `t.accept();` line to the text "current obj calling obj".

current obj  
calling obj

basic

app

req

```

int n1
n1=12
n1=55
    
```

Diagram for basic:

n1
66

Annotations:   
 - **n1\_ref** is written above the memory box.   
 - **5500** is written below the memory box.

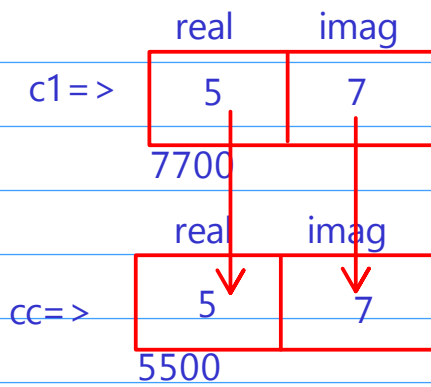
`int& ref = n1`

`ref=66`

`n1=> 66`

`ref=> 66`

`n=> int type vari`  
`int* p=> int pointer type vari`  
`int& r=> int ref type vari`



```
complex
{
pub:
sum(← complex& c2)
{
    this->real += c2.real
}
}
```

```
main()
{
    complex c1(7,6)
    complex c2(3,2)
    complex c3

    //c1.real+c2.real

    c1.sum(c2)
}
```