

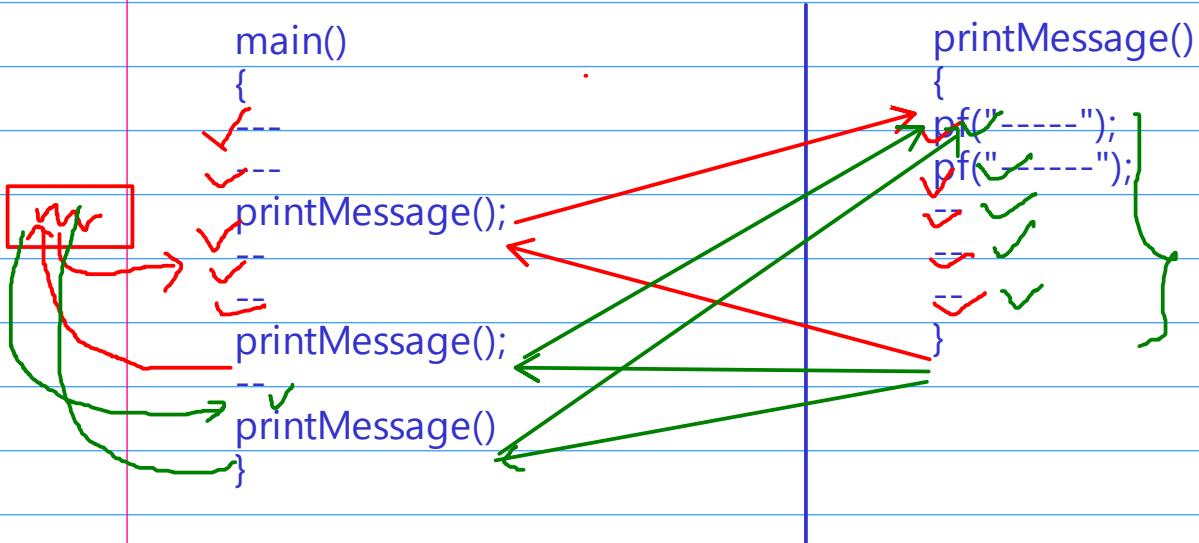
## C++ => C + OOP

C++ => 100%

35 % => sec B

70%

FAR



#\$

✓ ~~void printValue(int a) => printValue@int~~ -zprintValuei  
✓ ~~void printValue(int a,int b) => printValue@int,int~~ -zprintValueii  
✓ ~~void printValue(char a) => printValue@char~~  
✓ ~~void printValue(int a,char c) => printValue@int,char~~  
✓ ~~void printValue(char c,int a) => printValue@char,int~~

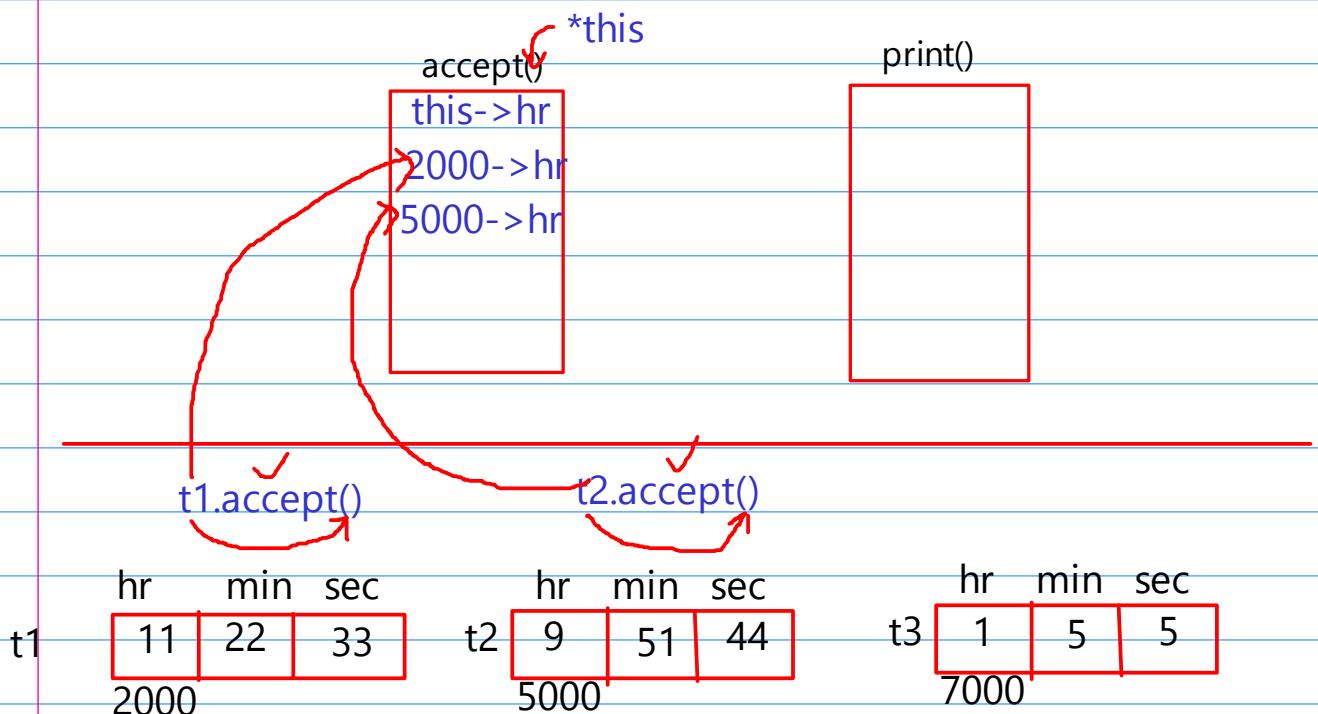
int -> 4 bytes => 32 bits

1 0 => 1 bit

bool => 1 byte => 8 bits

book  
 price ✓  
 auth ✓  
 name ✓  
 subj ✓  
 pages ✓  
 IDBI ✓  
 year ✓  
 versi ✓  
 salary ✓  
 roll\_No ✗

time  
 hr  
 min  
 sec



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

## Structure in C

```

struct time {
    int hr, min, sec;
};

gb void accept( struct time *p) {
    scanf("%d:%d:%d", &p->hr,
    &p->min, &p->sec); 2200->hr
} p=4400

Main()          hr min sec
{               11 22 33
    struct time t; 2200
    accept(&t);
}

```

## class in C++

```

class time { const 5500
    int hr, min, sec; time *this
    void accept();
mb scanf("%d:%d:%d",&hr, &min,
&sec); &this->hr
} 5500->hr
};

Main()          hr min sec
{               11 22 33
    time t; 5500
    t.accept();
}

```

current obj /  
calling obj

## basic

```

int n1;
n1=10
n1=15

```

**int &ref = n1**

ref = 50

cout<< ref => 50

cout<<n1 => 50

## app

n1	ref
50	
2200	

## req

int a => int data type vari as a

int \*p => int pointer type vari

int &r => int ref type vari

```

complex
{public:
    sum(complex &c2)
    {
        this-> c1
        c2 => para
    }
}

```

```

main()
{
    complex c1(5,7)
    complex c2(3,2)

    c1.real+c2.real //error
    c1.sum(c2)
}

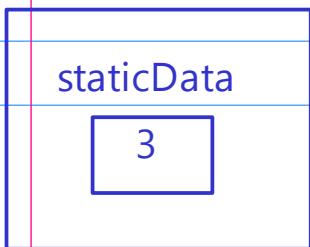
```

```

void staticDemo()
{
    int simple=1;
    static int staticData;
    staticData++;
}

```

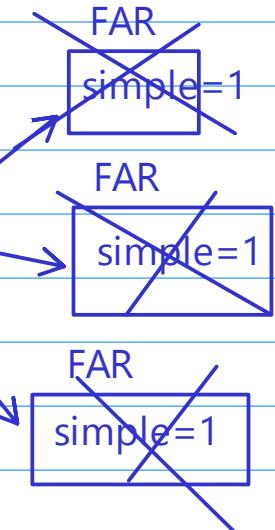
DS



```

main()
{
    staticDemo();
    staticDemo();
    staticDemo();
}

```



account

accNo

rate\_of\_intr

3.6

a[ 10000000 ]

10000000X4

4

c1=>

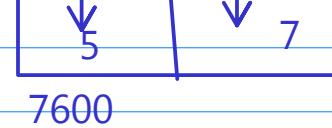


5600

shallow copy

old case

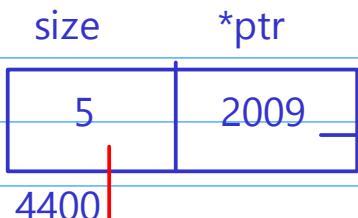
cc=>



new case

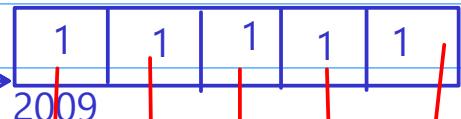
stack

a1=>

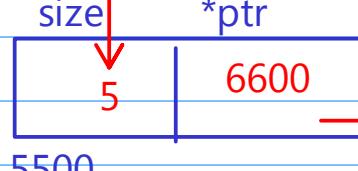


4400

heap



ac=>



5500

