

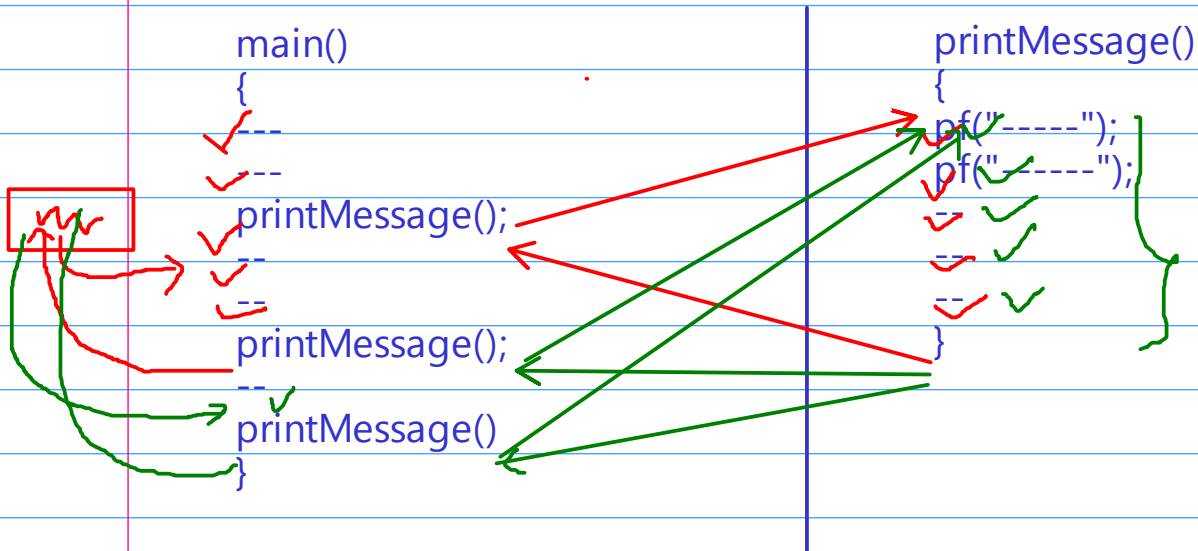
C++ => C + OOP

C++ => 100%

35 % => sec B

70%

FAR



#\$

~~void printValue(int a)~~ ¹ => printValue@int -zprintValue*i*
~~void printValue(int a,int b)~~ ² => printValue@int,int -zprintValue*ii*
~~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

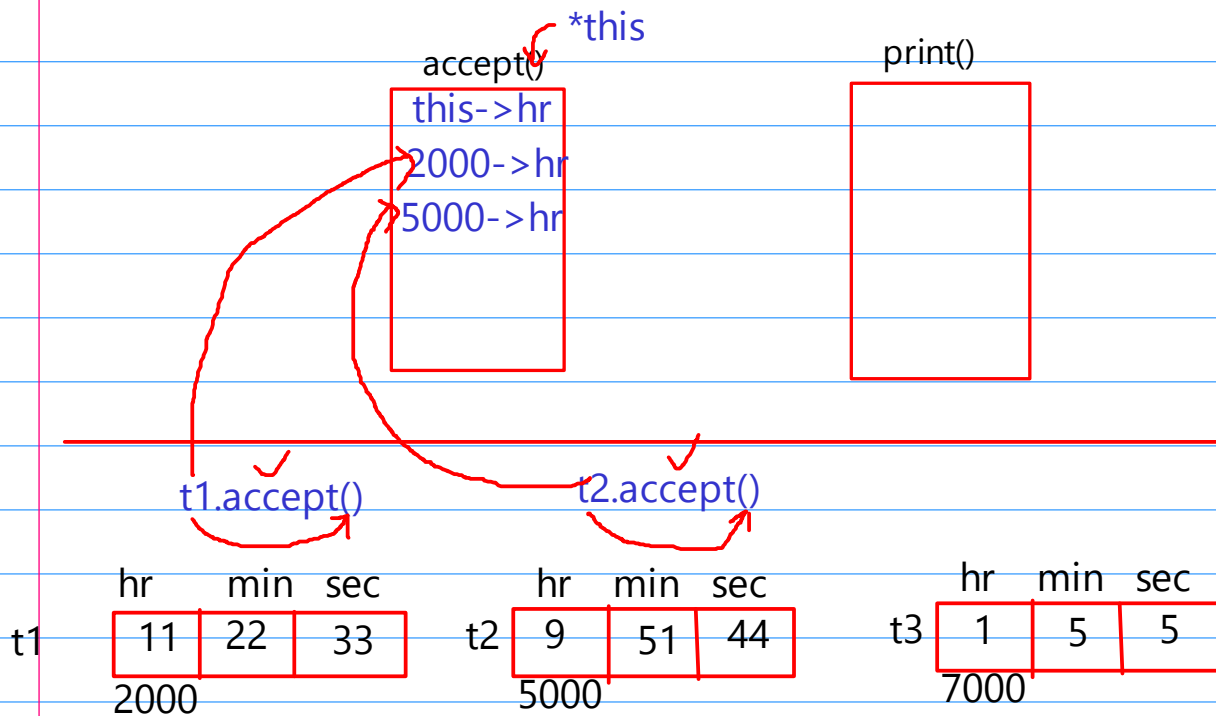
bool => 1 byte => 8 bits

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

class in C++

gb

```
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);
}
```

2200

2200->hr

p=4400

hr min sec

11 22 33

2200

```
class time {
    int hr, min, sec;
    void accept(){
        scanf("%d:%d:%d",&hr, &min,
        &sec);
    }
};

Main()
{
    time t;
    t.accept();
}
```

const 5500

time *this

&this->hr

5500->hr

hr min sec

11 22 33

5500

current obj /
calling obj

basic

app

req

```
int n1;
n1=10
n1=15
```

n1 ref

50

2200

```
int &ref = n1
```

```
ref = 50
cout<< ref => 50
cout<<n1 => 50
```

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)
}

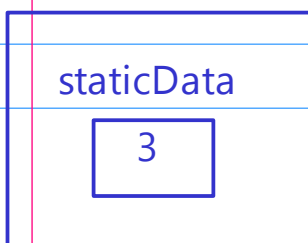
```

```

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

```

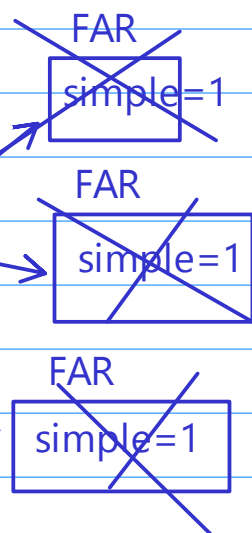
DS



```

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

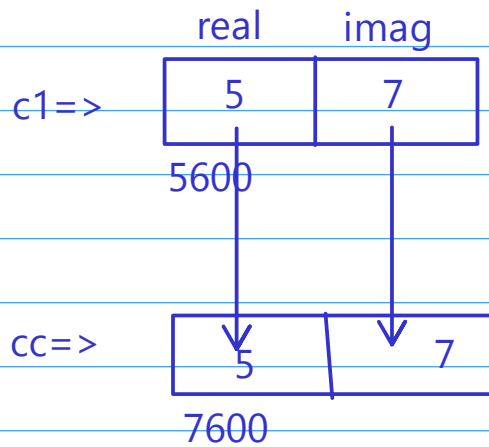
```



account
accNo
rate_of_intr
3.6

a[10000000]

10000000X4
4



shallow copy
old case

new case

