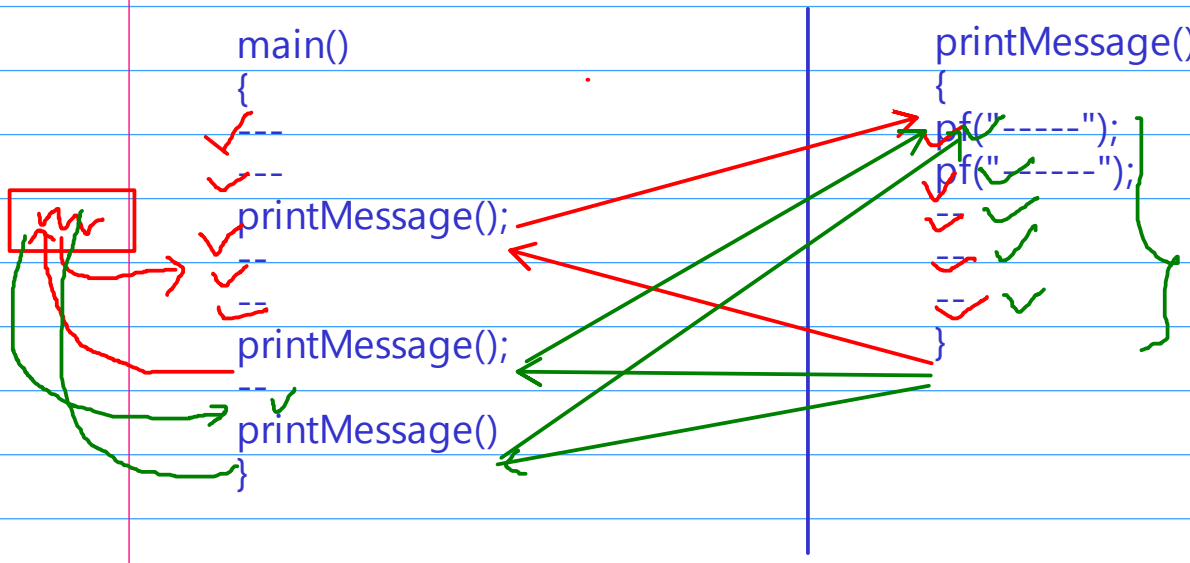
C++ => C + OOP

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

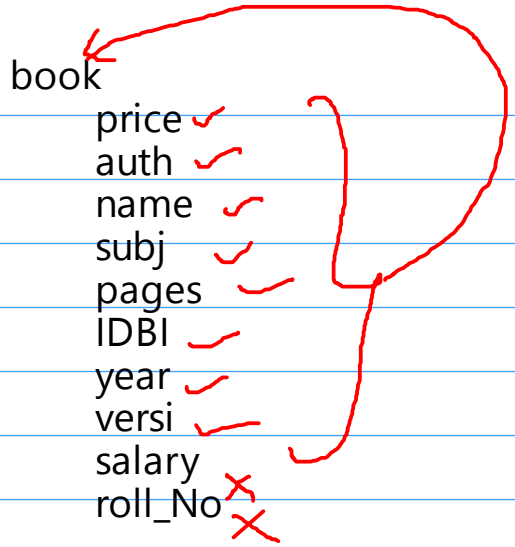
void printValue(int a,char c) => printValue@int,char

void printValue(char c,int a) => printValue@char,int

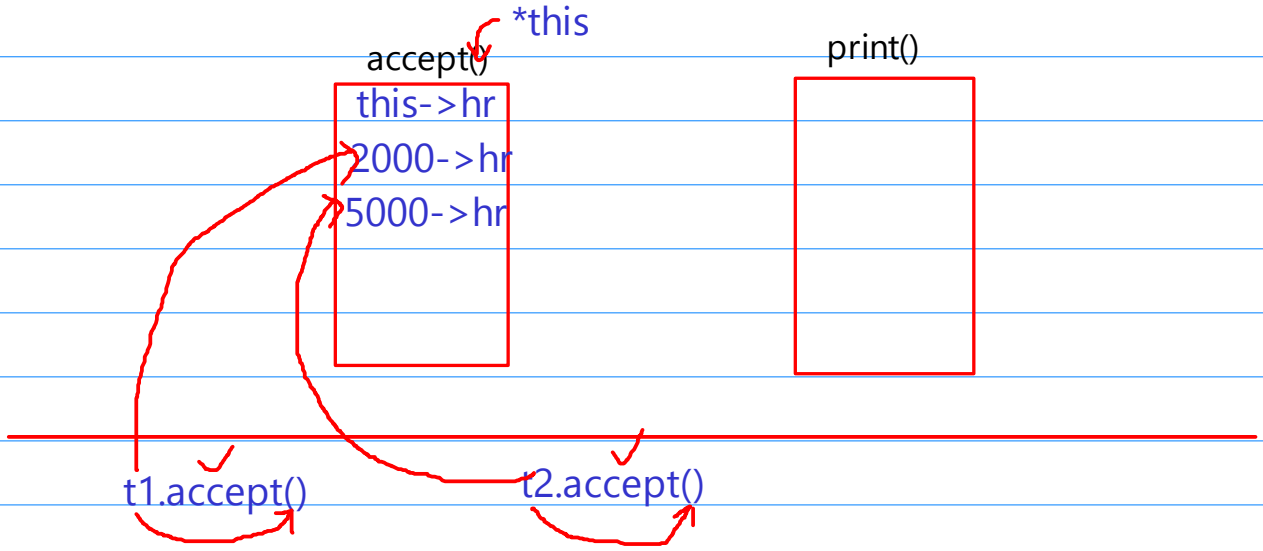
int -> 4 bytes => 32 bits

1 0 => 1 bit

bool => 1 byte => 8 bits



time
hr
min
sec



	hr	min	sec	
t1	11	22	33	2000
t2	9	51	44	5000
t3	1	5	5	7000

cpp => 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);
 }
```

2200
2200->hr
p=4400
hr min sec
11 22 33
2200

class in C++

```
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
t=> 11 22 33
5500

current obj /
calling obj

gb

mb

basic app req

int n1;
n1=10
n1=15
int &ref = n1
ref = 50
cout<< ref => 50
cout<<n1 => 50

n1 ref
50
2200

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

staticData

3

```
main()
{

      staticDemo();
      staticDemo();
      staticDemo();

}
```

FAR simple=1

FAR simple=1

FAR simple=1

account

accNo

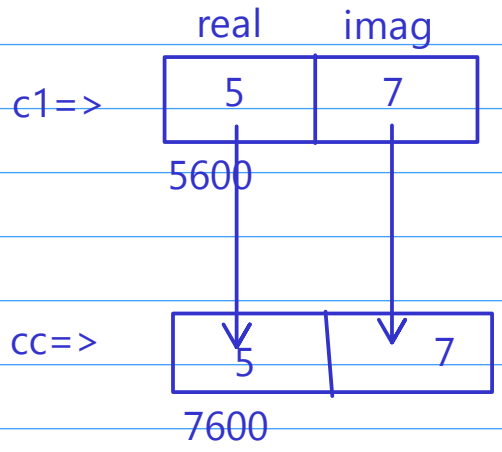
rate_of_intr

3.6

a[10000000]

10000000X4

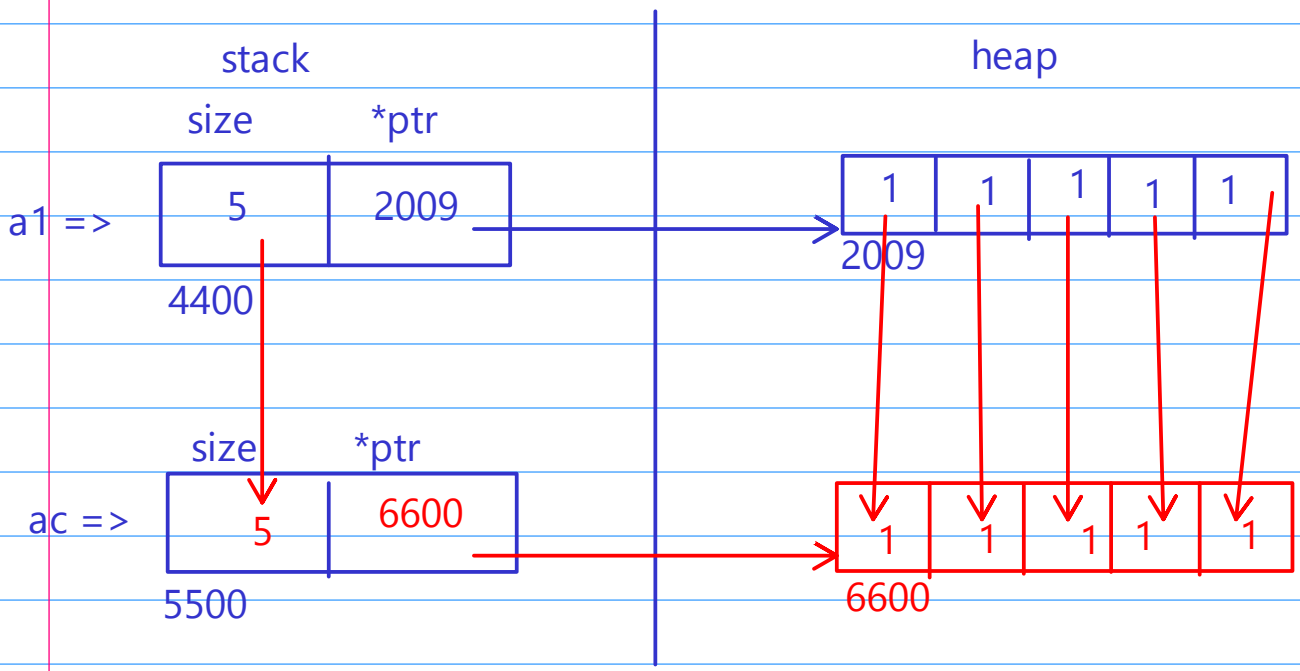
4

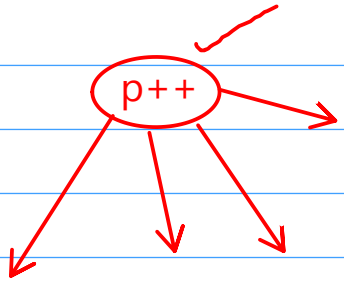


shallow copy

old case

new case





Abstraction

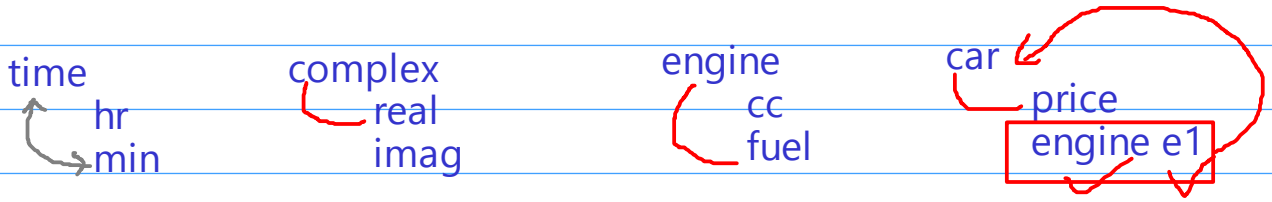
```
main()
{
printf("Enter %d",num);
        //calling fun
```

Encapsulation

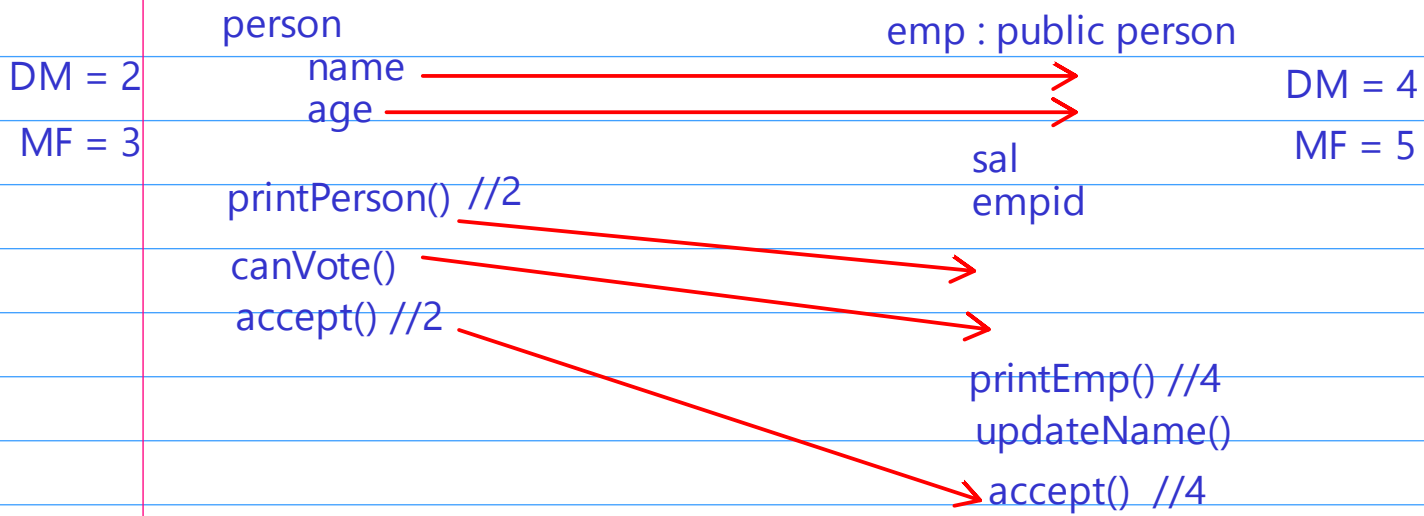
```
printf(-------)
{
--
--
--
--
--
--
--
}
```

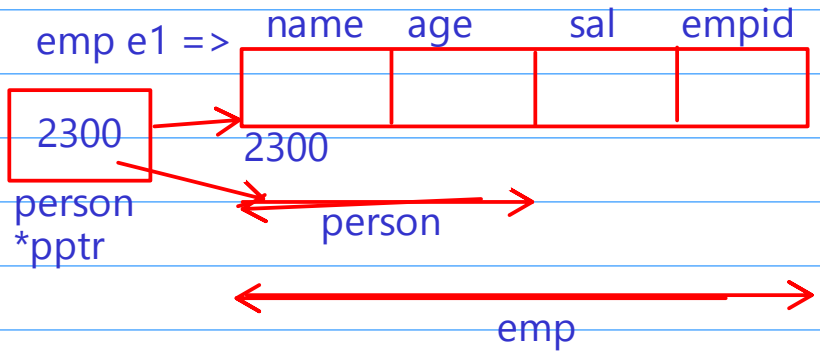
```
main()
{
account a1;
a1.deposite()
a1.withdrow()
```

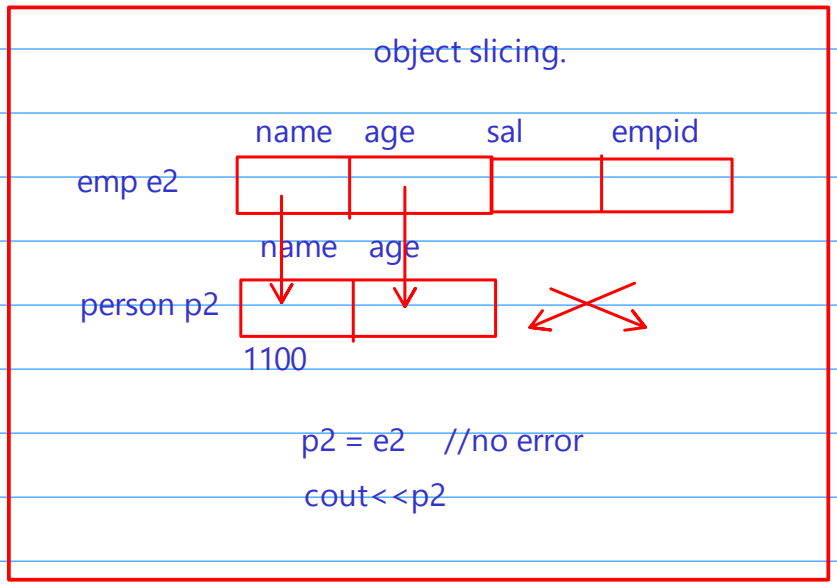
```
class account
{
private:
accNo, PAN
bal,
name
--
--
fun1(){--}
fun2() {--}
pub:
deposite(){--}
withdrow(){--}
}
```

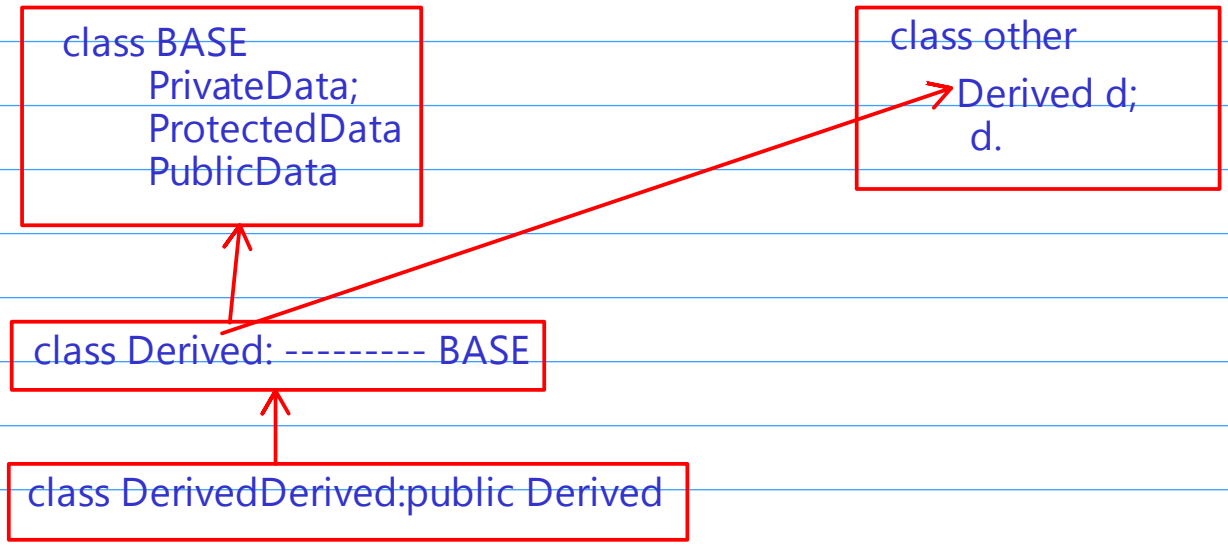


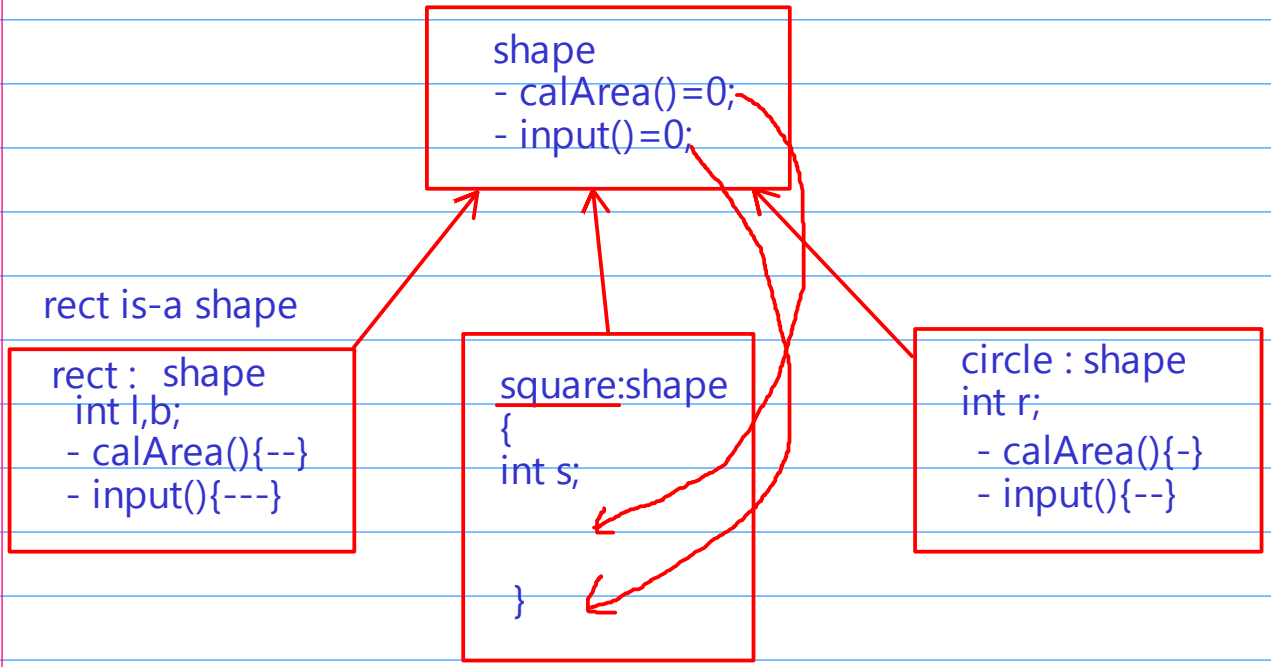
emp is-a person








class BASE
PrivateData;
ProtectedData
PublicData
class other
Derived d;
d.
class Derived: --------- BASE
class DerivedDerived:public Derived


shape
- calArea()=0;
- input()=0;
rect is-a shape
rect : shape
int l,b;
- calArea(){--}
- input(){---}
square:shape
{
int s;
}
circle : shape
int r;
- calArea(){-}
- input(){--}

base class functions

person :: bool canVote() => fully complete

person :: virtual void accept() => partially completed

shape :: virtual void calArea() = 0; => fully incomplete