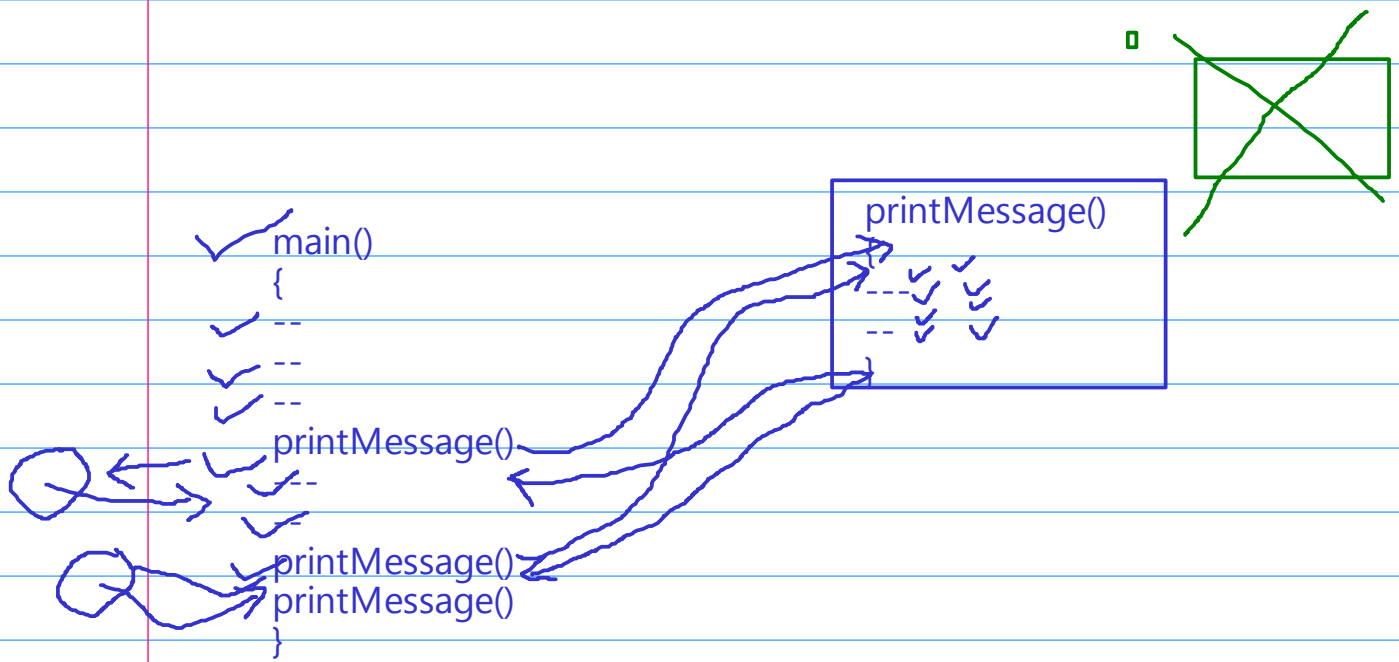
CDAC---> IT placement
CCAT --->rank
DAC
DMC
DBDA
DESD
Ditiss

section B
C
DS
OS
DCN
oop C++

lecture Lab

C++ --> oopl
2000+
C++ --->C+oop
100% --->35 -- 40 %

C++---> DAC-->100



void printValue(int n1) => printValue@int

void printValue(int n1,int n2) => printValue@int,int

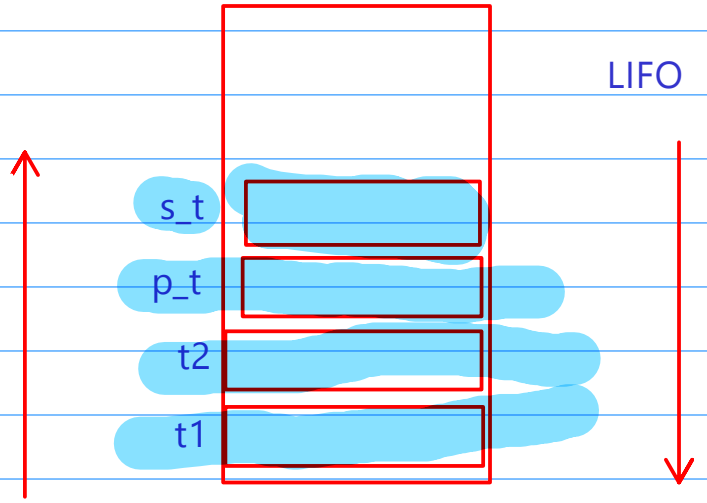
void printValue(char ch) => printValue@char

void printValue(int n1,char ch) => printValue@int,char 2

void printValue(char ch,int n1) => printValue@char,int 2

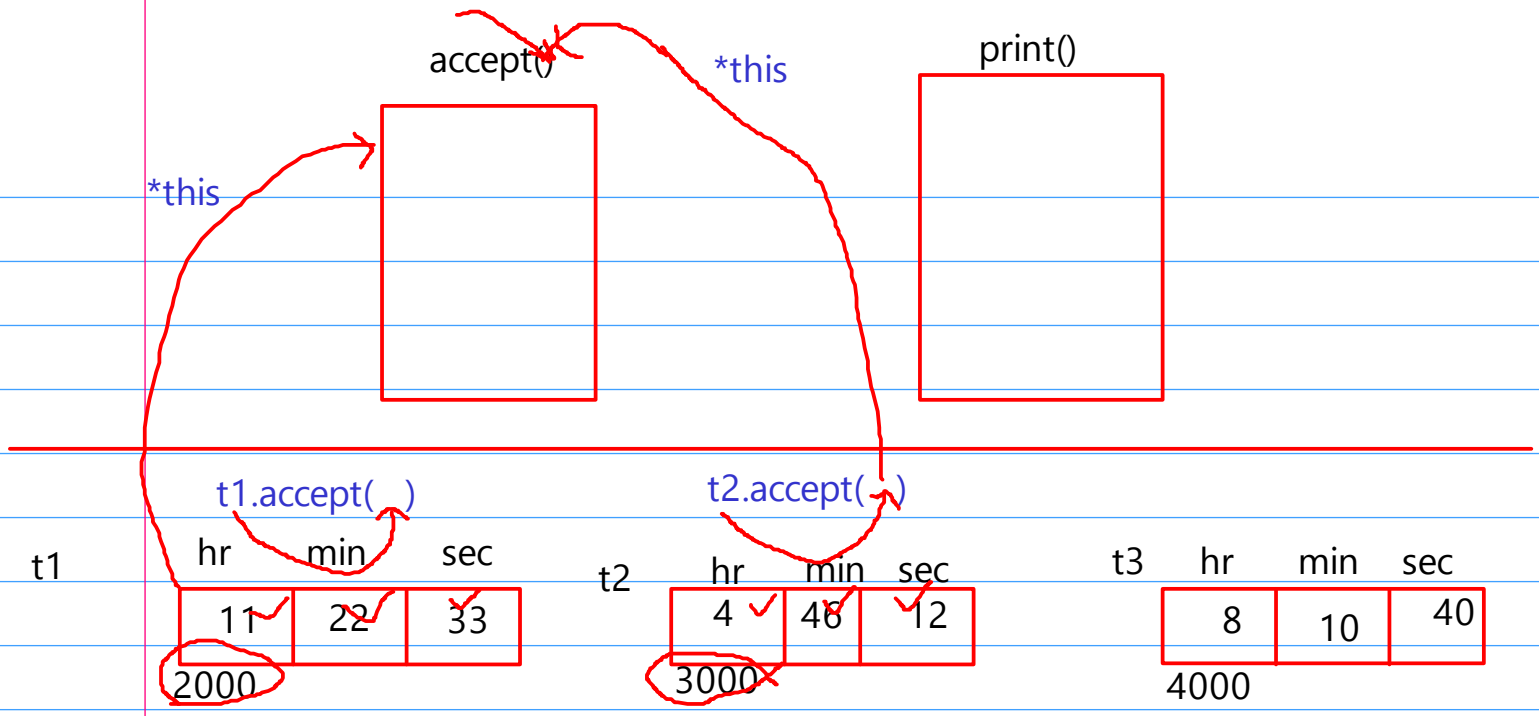
stud
-name
-age
-rollno
-sal
-mgr name

time
-hr
-min
-sec



7 --> 4th -- >3
5th --> oop ***
6th ---> inheritance
7th --->
feedback mid-1

sat -2
mon -2



struct 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);
 }
```

t: 11 | 22 | 33 — 5000

class in cpp

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

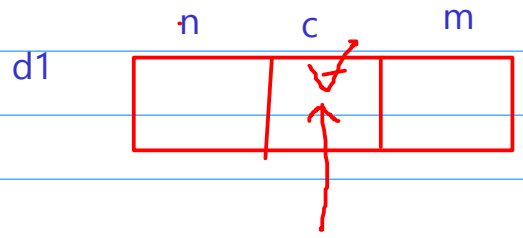
MF time *this

&this->hr

const

t: hr | min | sec — 9000

current obj/ calling obj

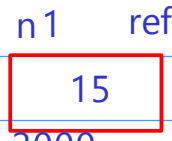


??

```
int num1;
num1=10
num1=55;

int num1=10
```

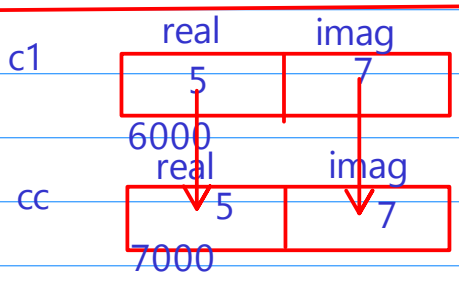
concept
use
req

int n1
n1=10
cout->10
int& ref=n1
ref=15
cout->n1=>15
cout->ref=>15



int& num1=ref
n1

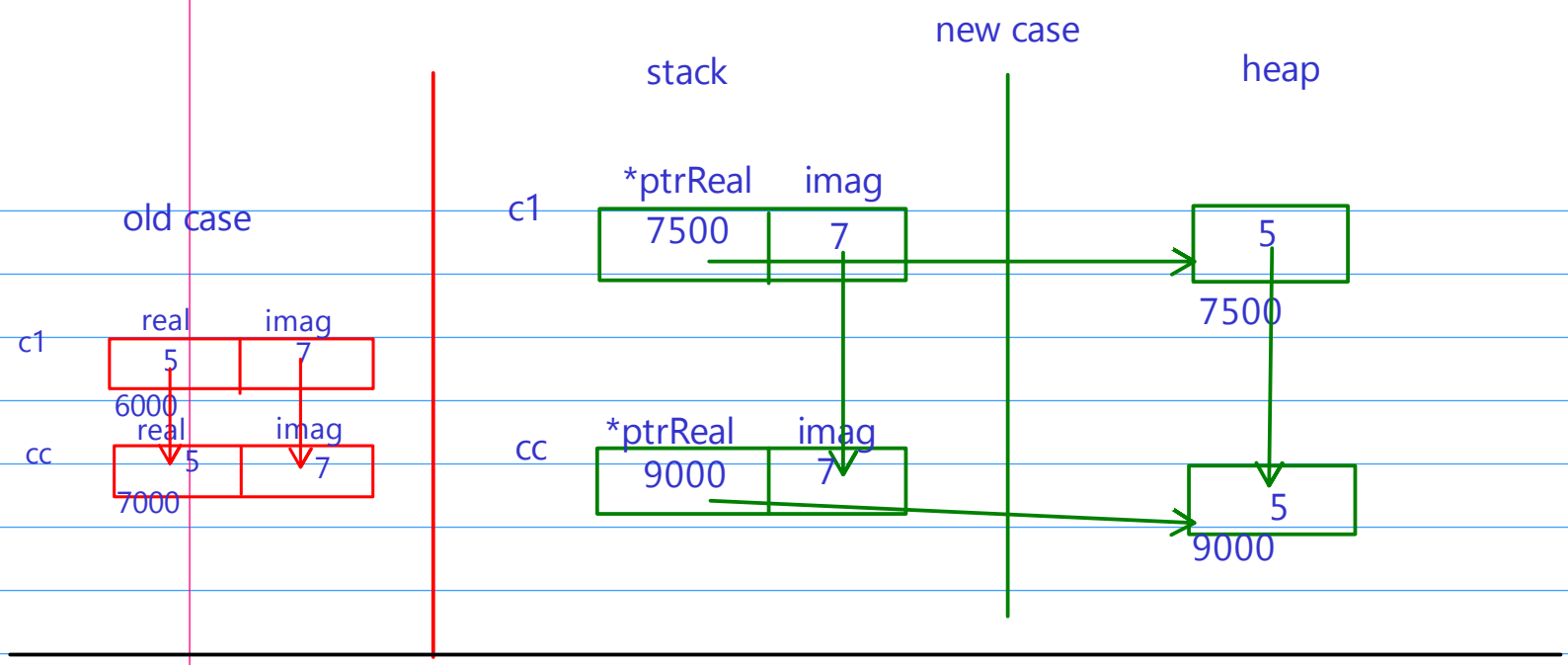
int n1-> int type data type
int* ptr-> int pointer type datatype
int& ref->int ref type datatype

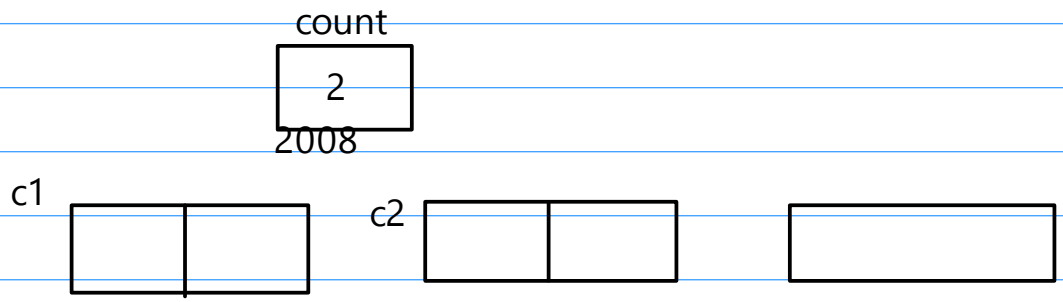



```
complex
{
public:
sum(complex& c2)
{
        complex c3;
c3.real=this->real+c2.real
        return c3
}

}
```

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

c1.real+c2.real

 //?
 c1.sum(  c2)
current obj

}
```

n1+n2





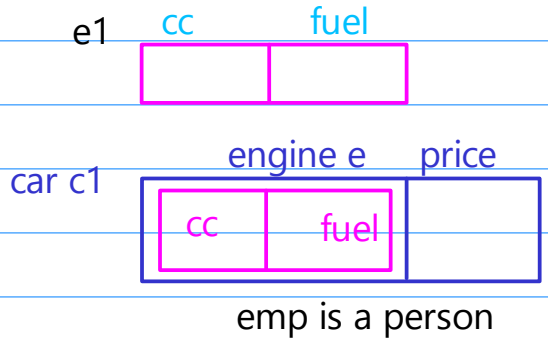
n1+n2; d1+d2 c1+c2

```
printf("enter  %d",n1);
```

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

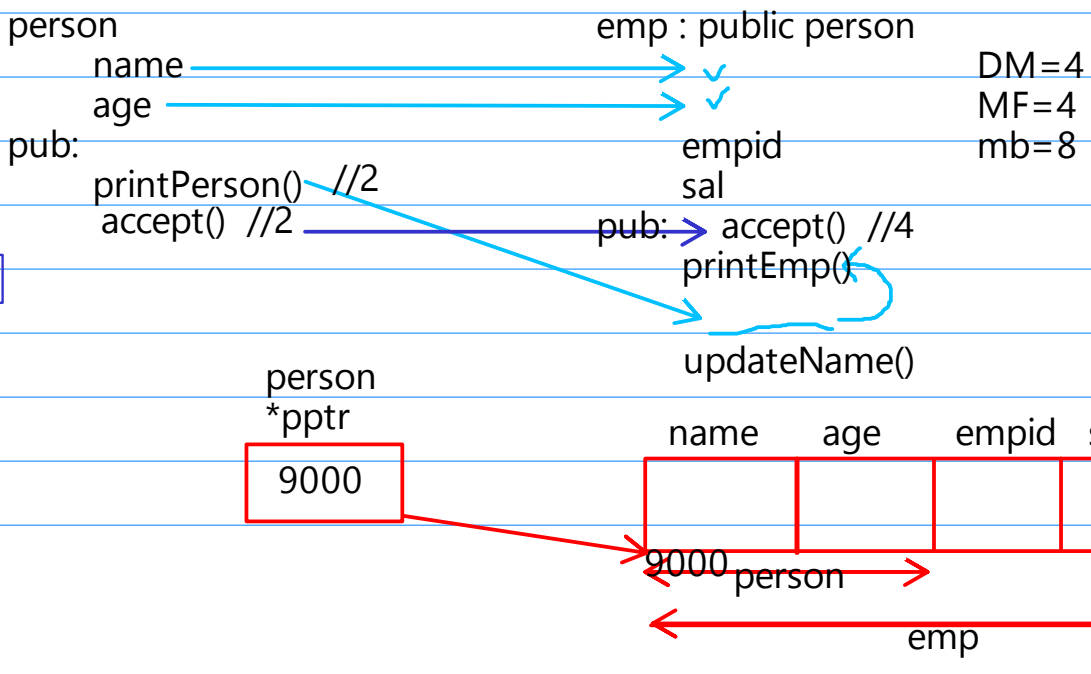
```
class accout
{
private:
--
--
--
fun()
fun2()
public:
withd()
disp()
}
```

```
main()
{
accout a1;
a1.withd()
a1.disp()

}
```



time
int hr
min
sec

complex
real
imag



s1=s2
p1=p2
c1=c2

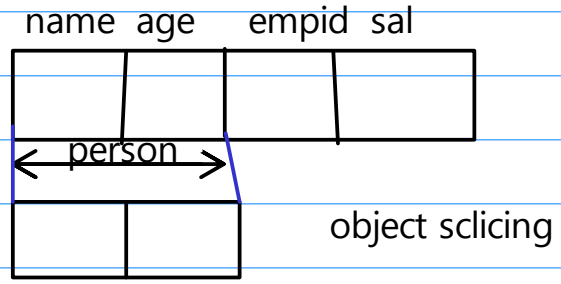
p1=e1

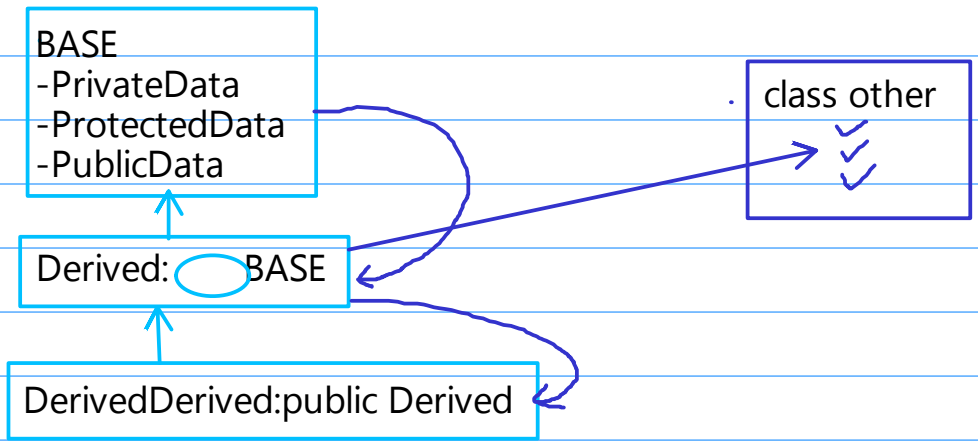
```
int  n1;        int* ptr;      ptr=&n1;
char ch;        char* chptr;   chptr=&ch;
ptr=&ch; //error
```

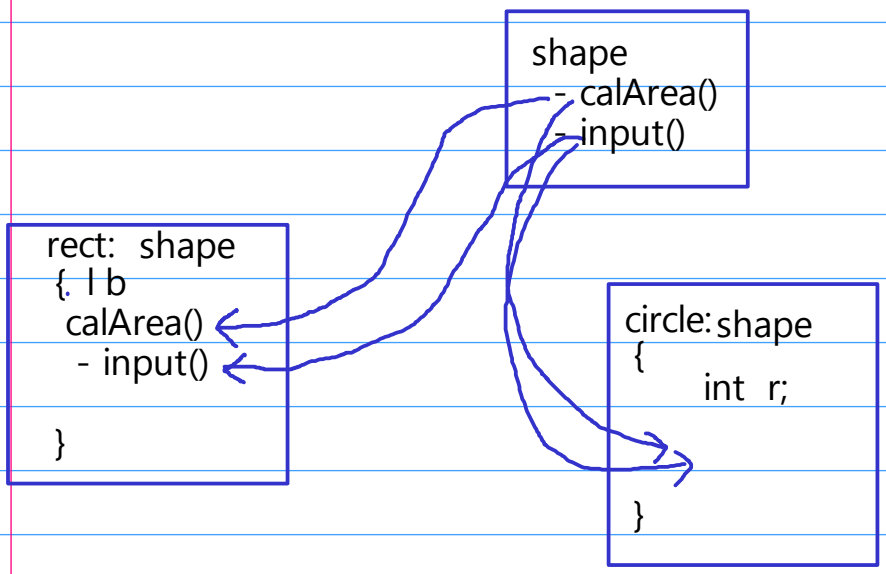
```
pesron p1    pesron* pptr;    pptr=&p1;
emp e1;      emp *eptr=NULL;  eptr=&e1;
```

```
complex c1;

complex c2(c1)    //copy

complex c3;

c3=c1; // =
```

person=emp
p1=e1







num1

20

num2

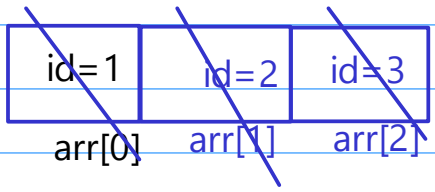
21

num3

20 21 20

count

3



1 2 3 3 2 1