

CDAC---> IT placement

CCAT --->rank

DAC

DMC

DBDA

DESD

Ditiss

section B

lecture Lab

C

DS

OS

DCN

oop C++

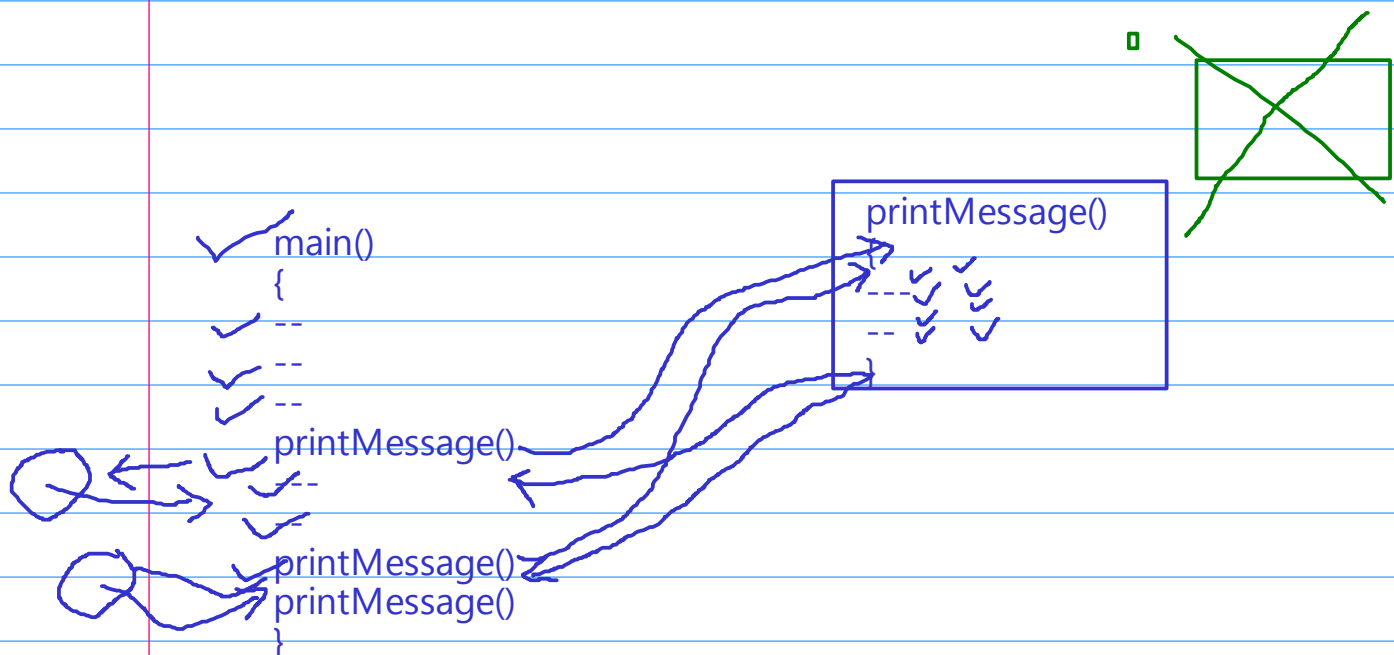
C++ --> oopl

2000+

C++ --->C+oop

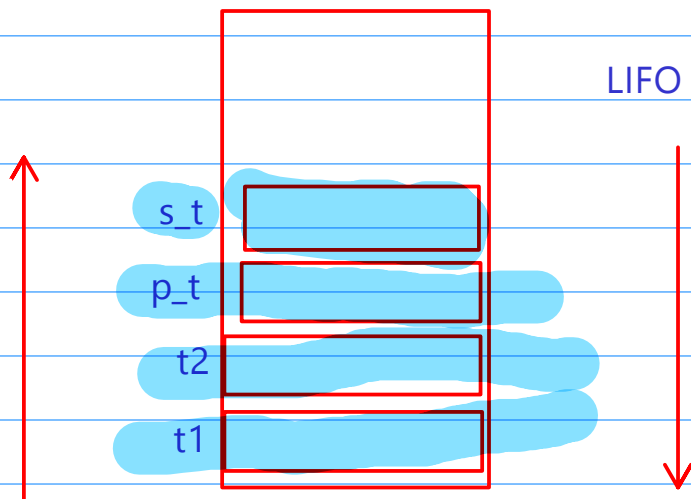
100% --->35 -- 40 %

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



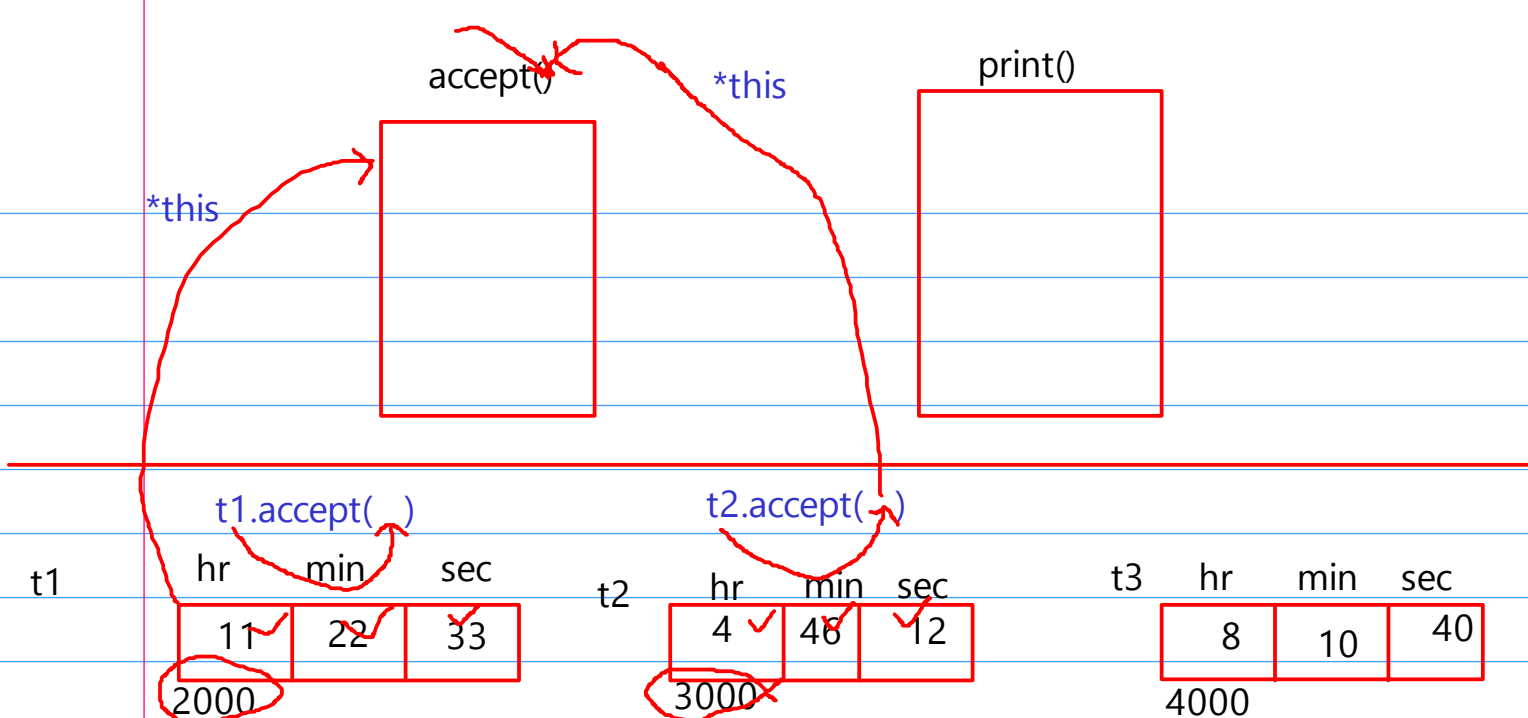
✓ void printValue(int n1) => printValue@int  
 ✓ void printValue(int n1, int n2) => printValue@int, int  
 void printValue(char ch) => printValue@char  
 ✓ void printValue(int n1, char ch) => printValue@int, char 2  
 ✓ void printValue(char ch, int n1) => printValue@char, int 2

stud	time
-name	-hr
-age	-min
-rollno	-sec
<del>-sal</del>	
<del>-mgr name</del>	



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

sat -2  
 mon -2



struct in c

class in cpp

```

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 showing the struct `time` with attributes `hr`, `min`, and `sec`. An object `t` is shown with values 11, 22, 33 and memory address 5000. Red arrows indicate the `accept(&t)` call.

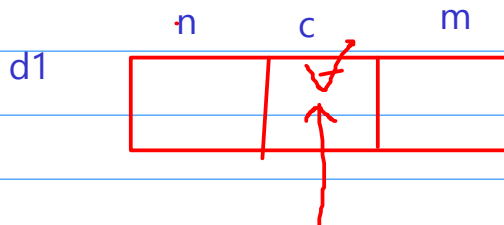
```

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

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

Diagram showing the class `time` with attributes `hr`, `min`, and `sec`. An object `t` is shown with values 11, 22, 33 and memory address 9000. Red arrows indicate the `t.accept()` call.

current obj/ calling obj



??

```

int num1;
num1=10;
num1=55;

int num1=10
  
```

Diagram showing the declaration and assignment of a variable `num1`. The initial value is 10, and it is updated to 55. A red arrow points to the final value 10.

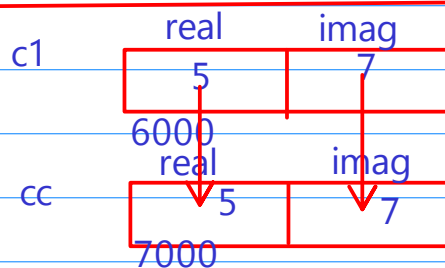
✓concept  
✓use  
✓req

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

n1    ref  
15    num1  
2000

int& num1=ref  
n1

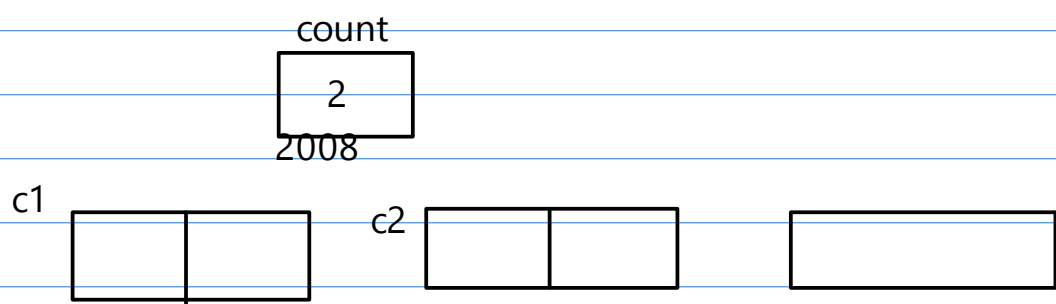
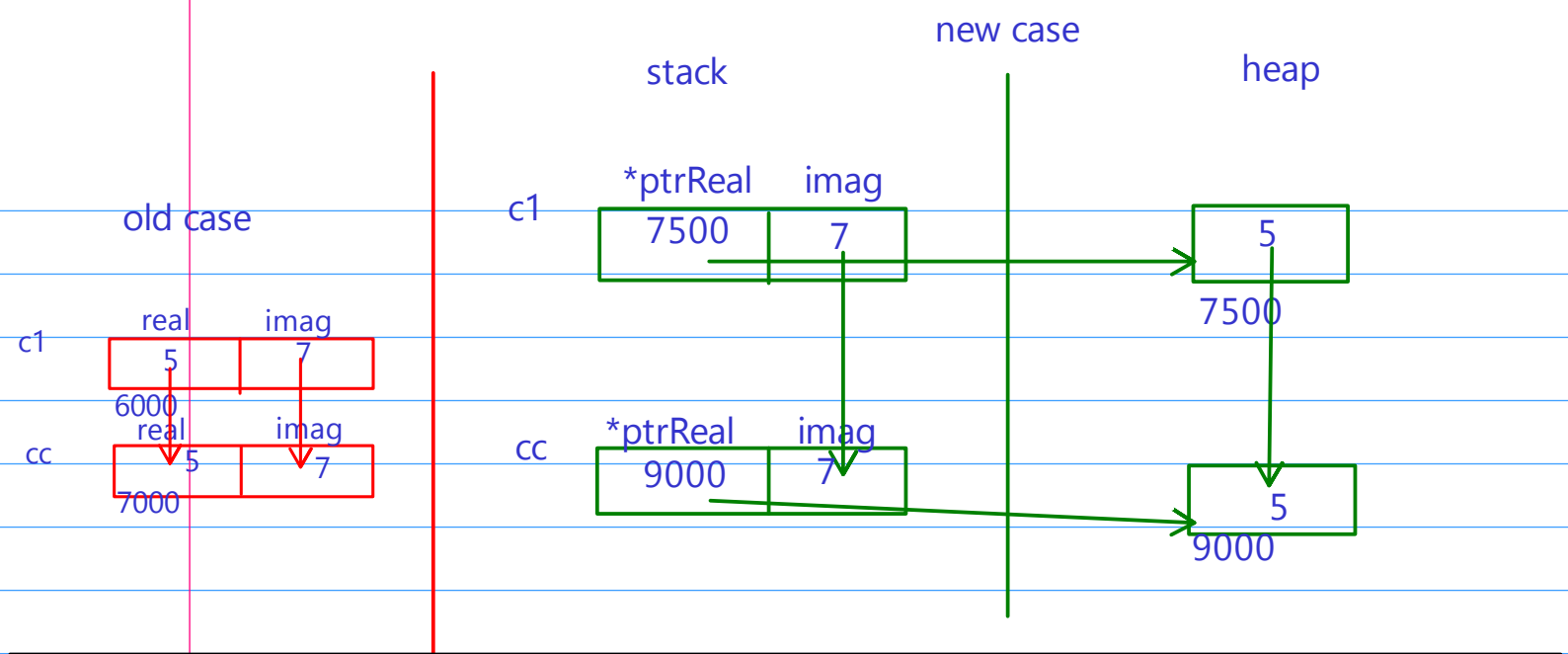
int n1-> int type data type  
int\* ptr-> int pointer type datatype  
int& ref->int ref type datatype



main() {    n1+n2

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

complex c1(7,6);  
complex c2(3,2);  
~~c1.real+c2.real~~  
//?  
current obj c1.sum( c2)



$n1+n2;$      $d1+d2$      $c1+c2$

printf("enter %d",n1);

printf(---)

class accout

{  
private:

--

--

--

fun()

fun2()

public:

withd()

disp()

}

main()

{

accout a1;

a1.withd()

a1.disp()

}

e1

cc

fuel



car c1

engine e price



emp is a person

time

int

hr

min

sec

complex

real

imag

person

name

age

pub:

printPerson() //2

accept() //2

emp : public person

empid

sal

pub: accept() //4

printEmp()

updateName()

DM=4

MF=4

mb=8



person

\*pptr

9000

name age empid sal



9000 person

emp

$s1=s2$   
 $p1=p2$   
 $c1=c2$

$p1=e1$

$\text{int } n1;$      $\text{int}^* \text{ptr};$      $\text{ptr}=\&n1;$   
 $\text{char } ch;$      $\text{char}^* \text{chptr};$      $\text{chptr}=\&ch;$   
 $\text{ptr}=\&ch;$  //error

$\text{pesron } p1$      $\text{pesron}^* \text{pptr};$      $\text{pptr}=\&p1;$   
 $\text{emp } e1;$      $\text{emp}^* \text{eptr}=\text{NULL};$      $\text{eptr}=\&e1;$

$\text{complex } c1;$

$\text{complex } c2(c1)$  //copy

$\text{complex } c3;$

$c3=c1;$  // =

$\text{person}=\text{emp}$   
 $p1=e1$

