C
DS
Apti
DCN
OOpc c++   ==>B
BD and AI

DAC --> C++

Lecture   Lab

7

oop --> 2000+ -->C++

C --> C++ -->OOP

```
int main()
{
    //--
    //--
    //---
    printMessage();
    //--
    //--
    printMessage();
    //-
    printMessage();
    printMessage();
    printMessage();
    return 0;
}
```

FAR

```
void printMessage()
{
    printf("\n Good morning ....");
    printf("\n Hello .. OM46 ... :)");
}
```

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

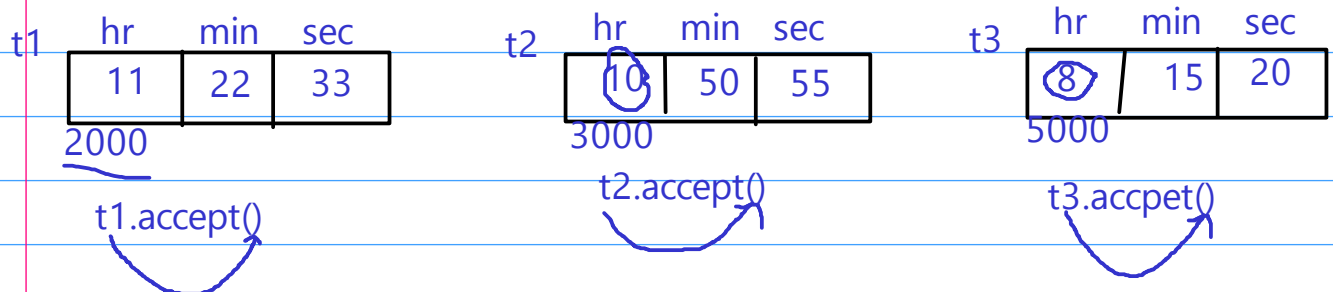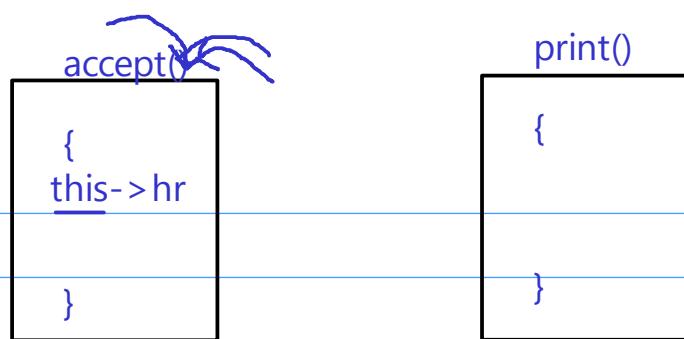emp   time
    name   hr ✓
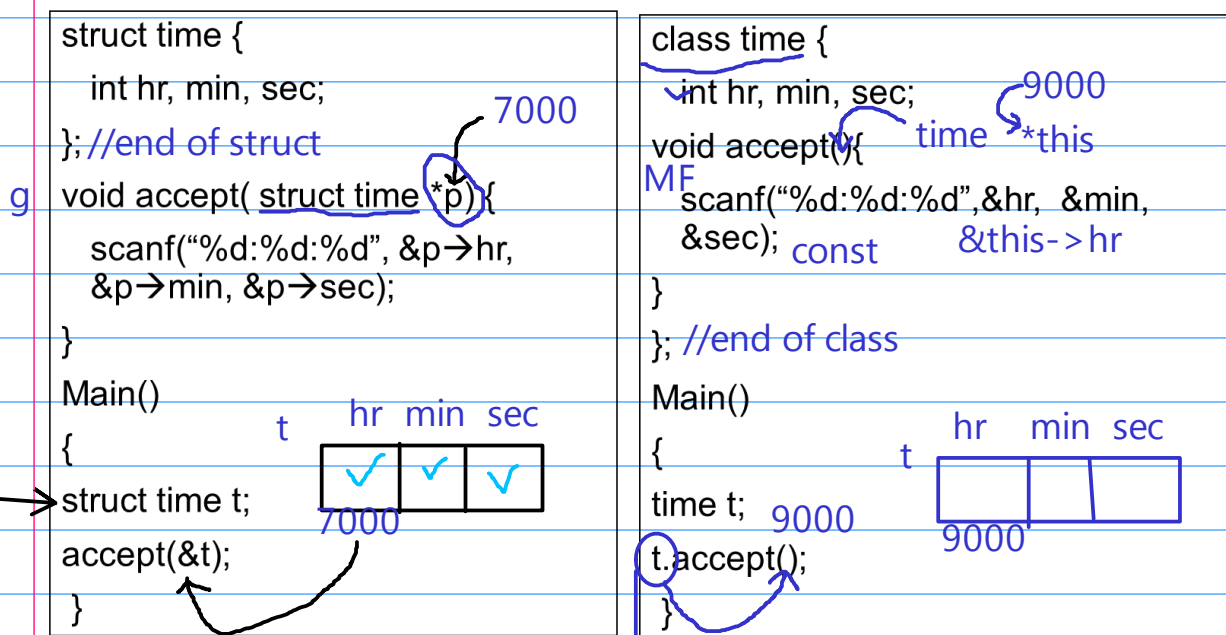   empid   min ✓
    sal   sec ✓
    age
    mgr
    roll_no

accept()

```
{
    this->hr

}
```

print()

```
{

}
```

---

t1

hr	min	sec
11	22	33

2000

t1.accept()

t2

hr	min	sec
10	50	55

3000

t2.accept()

t3

hr	min	sec
8	15	20

5000

t3.accpet()

this -> java, c++, c#
self -> py, objC

C

```
struct time {
    int hr, min, sec;
};  //end of struct
g   void accept( struct time *p ){        7000
        scanf("%d:%d:%d", &p→hr,
        &p→min, &p→sec);
    }
    Main()
    {
        struct time t;
        accept(&t);
    }
```

t

hr	min	sec
✓	✓	✓

7000

C++

```
class time {
    int hr, min, sec;            9000
    void accept(){    time  *this
MF      scanf("%d:%d:%d",&hr,  &min,
        &sec);   const     &this->hr
    }
};  //end of class
    Main()
    {
        time t;      9000
        t.accept();
    }
```

t

hr	min	sec

9000

current/ calling obj

stack LIFO

```
┌─────────────┐
│    t_s      │
├─────────────┤
│    t_p      │
├─────────────┤
│    t2       │
├─────────────┤
│    t1       │
└─────────────┘
```

```
        n    c    m
┌────┐  ┌────┬────┬────┐
│ 22 │  │    │    │    │
└────┘  └────┴────┴────┘
```

day 4
f/b
oop_cpp_quiz-1
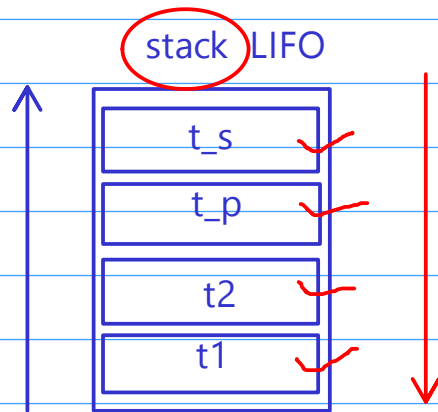
ref
new - delete
deep - shallow

3
1:concept ✓
2:app ✓
3:req ✓

```
int n1;
n1=5
n1=10
```
           n1   ref
        ┌─────────┐
        │   15   │
        └─────────┘
         5000

```
int& ref=n1;
```
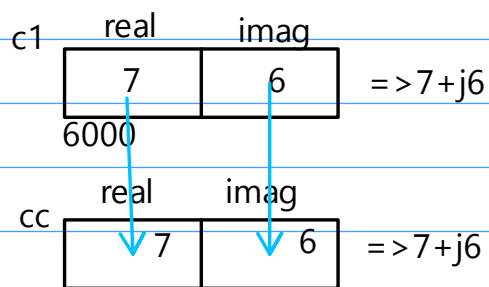```
ref=15
  cout->n1==>15
  cout->ref==>15
```

int -> interger type data type
int* -> interger pointer type datatype
int& -> interger ref type datatype

```
// int a1=a
//int& b1=b
void mySwap(int a1,int& b1)     int main()
{                               {
    int temp = a1;                  int a=11,b=99;
    a1=b1;                          cout<<"\n before swap a="<<a<<" b="<<b
    b1=temp;                        mySwap (a,b);
}                                   cout<<"\n after swap a="<<a<<" b="<<b;
                                }
```

by ref

by value

c1

real	imag
7	6

=>7+j6

6000

cc

real	imag
7	6

=>7+j6

n3=n1+n2

```
class complex
{
public:
sum(  complex& c2)
{       7
this->real+c2.real;
this->imag+c2.imag


}


}
```

```
main()
{
complex c1(7,6)
complex c2(3,2)
complex c3;

c3 = c1.sum( c2)


}
```

New case deep

old case
shallow

stack

heap

c1

real	imag
7	6

=>7+j6

6000

cc

real	imag
7	6

=>7+j6

c1

*ptrReal	imag
7500	6

7

7500

cc

*ptrReal	imag
5000	6

7

5000
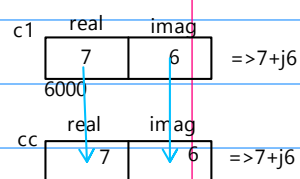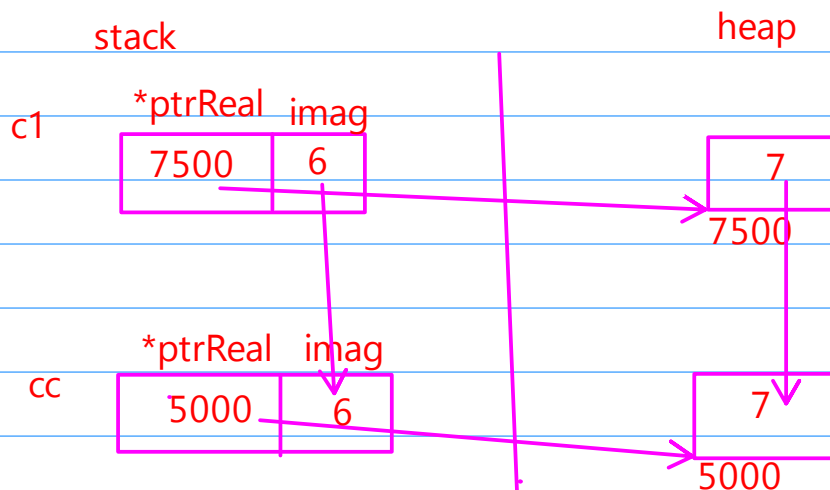
6=>10Q   1=20Q
5th -> oops
6th -> inheritance
7th -> adv, exp,temp,mcq

value of count=0   address of count=0x4e2008

n1 + - n2

t1+t2   c1+c2

p++ ✓

abstraction   encapsulation

printf("hello %d %c",n1,ch);   printf(......)
                                      {
                                      --
                                      --
                                      --

                                      --
                                      }

encapsulation

class account   main()
{   account a;
private: a.--   abstraction
    --   a.--
--
--
fun(){}
fun2(){}
public:
withd()
credit()
}

47--> engine e1;   43 --> car c1

e1   cc   fuel   price   engine e

2000   5000   cc   fuel