OOP Course

Created by : Israa Abdelghany

LinkedIn: www.linkedin.com/in/israa-abdelghany-4872b0222 GitHub: https://github.com/IsraaAbdelghany9

Lecture 2

```
struct in c & C++ => functions & access modifier (default public)

Class => default private => fill now its the only diffrence

Void print (emp e) => called stand alone function => can be called

E cout << e.get | d() << ":" << e.get Name () directly

e.set | d (9000) i => Call by value
```

functions must be after class Estruct & include
instant member or function member - functions of classes

```
void print (emp e);
      41
          int main()
     42
     43
              el.setId(10);
     45
              el.setName("aly");
     46
             el.setAge(30);
     47
             print2 (&e1); => creates a copy in function
     48
        //stand-alone function
        void print (emp e) //
              cout<<e.getId()<<":"<<e.getName()<<":"<<e.getAge();</pre>
             e.setId(9000); = will be deleted after exiting the function
                                      because it's a copy
         void print2 (empt e)
     59
             cout<<e->getId()<<":"<<e->getName()<<":"<<e->getAge();
              e -> settd (9000); will save it even after exiting
Logs & others
```

* any member function has a hidden parameter called this pointing to its object

** if you debug the Code you will find this

** this -> id be cause this a pointer

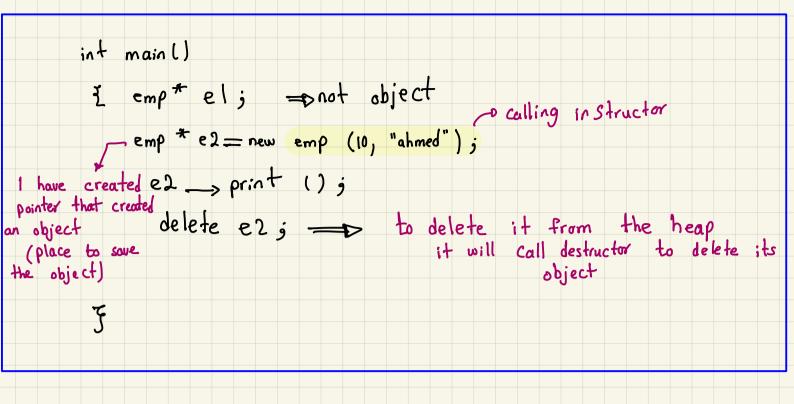
```
class Complex
     int real;
      Public: void
// setters & getters
              void set Real (int_real)

Treal = _real; 3
               Void set img (int_img)

img = img; 5
                int get Real () = return real; 5
                int geting ()
{return img; }
                  void print ()
                  [ if (img 70)
                       cout << real << "+" << img << "j";
                    else if limg Lo)
                         cout << real << ing << "j" j
                        else
                         Cout << real ;
                    3
                  void set Complex (int _real , int _ img)
                         I real = real j
                            img = _ img;
```

```
by it's this 7
            Complex Add (Complex c)
                Complex res ;
                 res real = c real + real;
                 res. img = c. img + img;
                 return res;
                // if I typed img = 9000; it will assign this value to
 int main ()
  E complex C1, C2, C3;
  C3 = C2. Add (C1);
                 o Calling by
                          value
calling
by address
 in function its
  called by this
 * Note :-
     polymor phisum - differ different functions by number of
                             arguments or type of them (overloading)
   * over loading depends on arguments not the return type.
   * constructor is called directly after the object creation
        it is used if we need to do something right after the object
     * constractor I is called once
                 2 has same name as the class
                 31 no return type
     emp () { age = 30 ;
                 id = 0;
strcpy (name, "no name");
```

```
Constructor because it don't
* constructor is called parameterless
   have parameters to get emp ()
 * we can overload the constructor.
 * emp el ez; -o will call the parameterless constructor
          e3 (log "aly", 30) of emp (int _id , char _name[], int _age)
                                              1 id = _id;
                                                  age = - age;
                                                  Stropy (name, -name);
          1 types just emp elez; No error
           I typed emp e 3 (10, "aly", 30); = with
                           void print()(
    cout<<endl<<iid<<":"<<name<<":"<<age;</pre>
                          emp(int _id, char _name[]) {
  id=_id;
                           strcpy(name,_name);
age=30;
                         emp(int _id, char _name[], int _age) {
  id=_id;
                          strcpy(name, _name);
age=_age;
                     53 your print(emp e);
55 void print2(emp* e);
                       int main()
                          emp e1, e2;
  * constructor can be private
  * when you create any constructors the default const. needs to be created
            too or error if you call it or emp els
  * destructor is ~ name_of_class
  * destructor is called when object is deleted or
                                                                Code is finished
   x if the function the objects created in is not ended
                                                                the object con't
                   be deleted because it's soved in stack
```



* Copy Constructor

used in 4 cases

search

* Creating object from object a

* Creating object from object call the copy constructor.

Exam: = mcq. 2 questrons code typing. LAB □ Complex _ setter, getter print constractor Add (member function & Stand alone) Demp class ____ constructor destractor trace