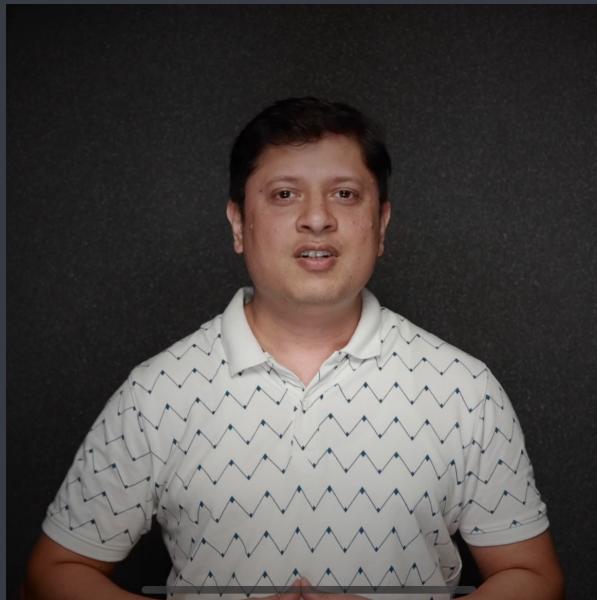


A job ready bootcamp in C++, DSA and IOT

## Constructor



Saurabh Shukla (MySirG)

# Agenda

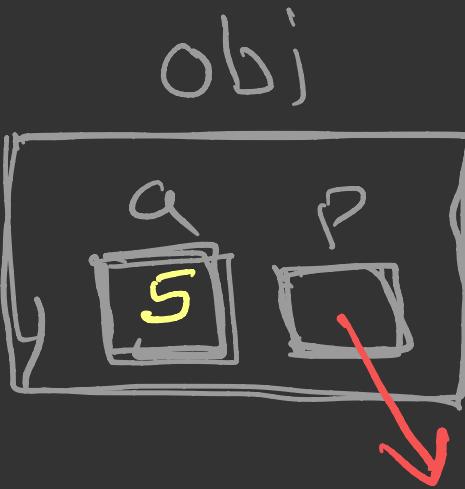
- ① Constructor
- ② Parameterized Constructor
- ③ Constructor Overloading
- ④ Default Constructor
- ⑤ Copy Constructor
- ⑥ shallow copy vs Deep Copy

## Constructor

- Constructor is a special member of the class whose name is same as the name of the class
- Constructor has no return type
- Constructor is invoked at the time of object creation (automatically)
- Constructor is an instance member
- Usually constructor is defined as public member but it can be private also.

- Programmer has to define Constructor, so he can write any code but it is useful to initialize properties of an object.

obj. setData(3,4);  
obj. updateData(5,7);



7

void updateData( int x, int y)

{

a=x;

\*p=y;

}

curr -- (int x,int y)

{

a=x;

p=(int\*)malloc(4);

\*p=y;

}

# Parameterized Constructor

- You can make a constructor with arguments.
- Constructor arguments are passed at the object creation.

## Constructor Overloading

- Programmer can provide multiple constructors in the class with different signatures.

## Default Constructor

- When programmer doesn't provide explicit constructor in the class, compiler creates an empty body, no argument constructor in the class

## Copy Constructor

- Either programmer has to provide copy constructor in the class or compiler itself provides copy constructor.
- Copy constructor is invoked for newly created object which is initialized with the object of the same class
- Formal argument of copy constructor must be a reference variable of same class

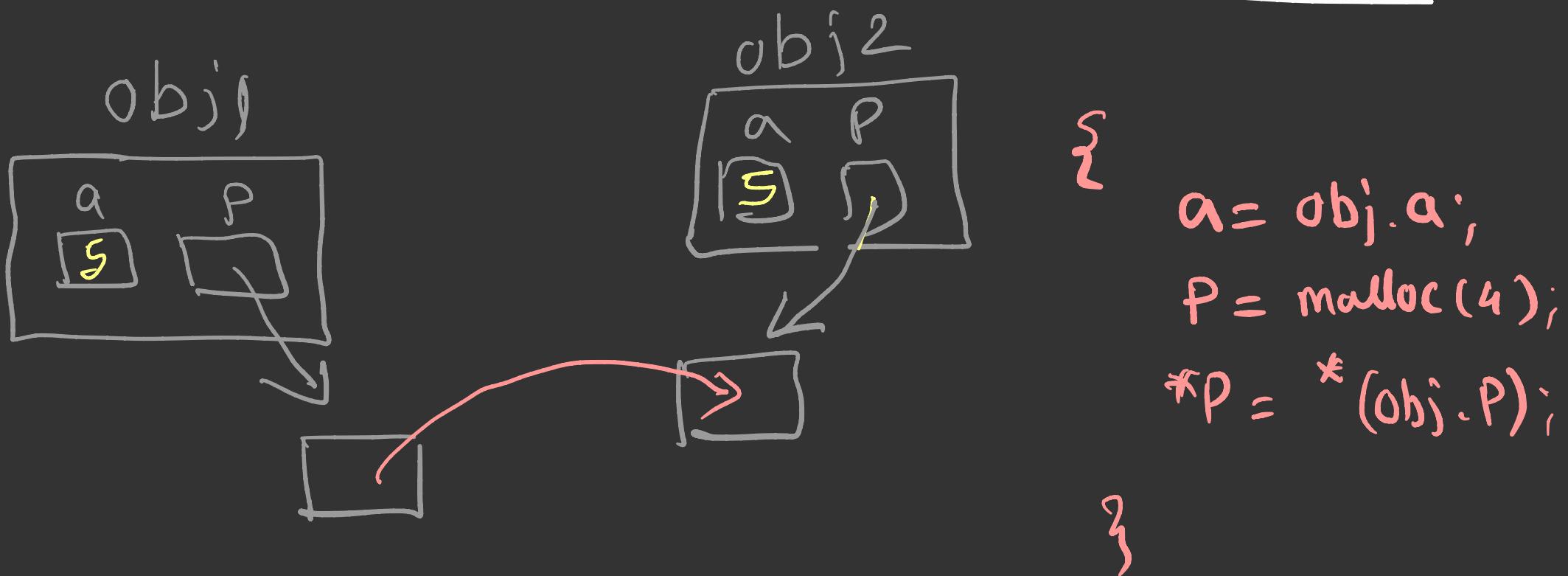
① when there is no explicit constructor defined in the class Compiler defines two constructors

- default constructor
- copy constructor

② when there is at least one explicit constructor of any type Compiler doesn't provide default constructor

③ when there is explicit copy constructor in the class compiler defines none

# Shallow Copy vs Deep Copy



X `obj2 = obj1;`