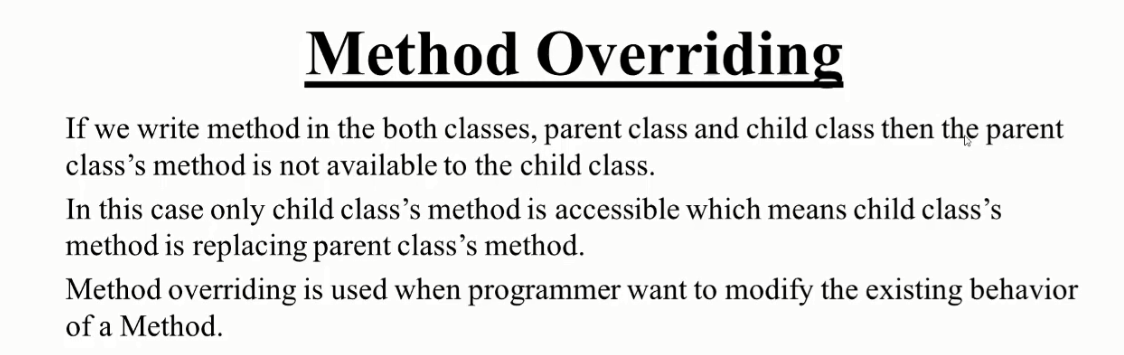
# Overriding, Overloading

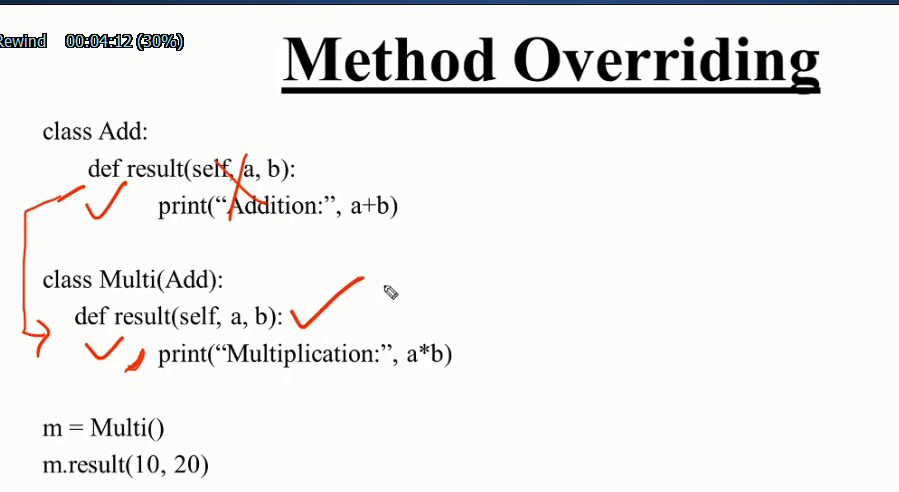
Operator overloading

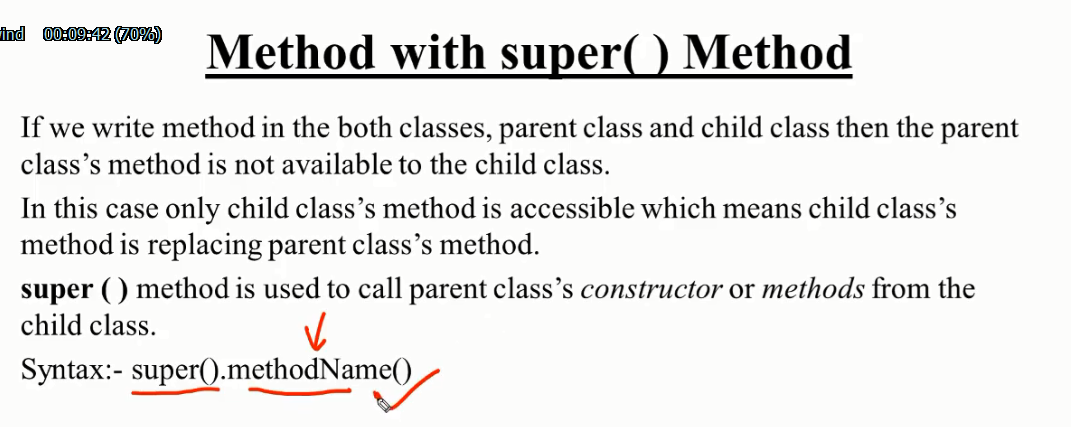
Function overloading

Method overloading

## Overriding







### Super use:

### Alter super formula use :

*#use super alter formula in method overriding*

class one:

**def** \_\_init\_\_(*self*):

        print('i am init method 1')

**def** fun(*self*):

        print('i am fun1')

class two:

**def** \_\_init\_\_(*self*):

        print('i am init metod 2')

**def** fun(*self*):

        print('i am fun2')

class three(*one*,*two*):

**def** \_\_init\_\_(*self*):

        one.\_\_init\_\_(*self*)

        two.\_\_init\_\_(*self*)

        print('i am init method 3')

**def** fun(*self*):

        one.fun(*self*)

        two.fun(*self*)

        print('i am fun3')

lol**=**three()

lol.fun()

## Overloading

Actually method overloading is not possible in python programming. er poreo bisoy ta Kemon hoy tar ekta example deya hoto

class one:

**def** fun(*self*):

        print('sum')

**def** fun(*self*,**x**):

*self*.x**=**x

        print('mutipication:',x)

sos**=**one()

sos.fun()

sos.fun(2)

on the above we seen that two method defined in a class both method name are same but difference in its parameters. Method one have not any extra parameter but method two have an extra parameter of x. its overloading.