

Ques. What is Decorators?

- A decorator is a design pattern in Python that allows a user to add new functionality to an existing object without modifying its structure. Decorators are usually called before the definition of a function you want to decorate.
- Decorators are used to add some design patterns to a function without changing its structure.
- A decorator function is a function that accepts a function as parameter and return a function(decorator ek function hai jo as a argument leta bhi function hai and return bhi function karta hai).
- Decorators allow us to wrap another function in order to extend the behaviour of the wrapped function, without permanently modifying it.

```
def decor(fun):  
    def inner():  
        print('befor inhance')  
        fun()  
        print('after inhance')  
    return inner  
  
def num():  
    print('hello mohit')  
  
result = decor(num)  
result()
```

Output:-
befor inhance
hello mohit
after inhance

- 2nd Type

```
def decor(fun1):  
    def inner():  
        print('befor inhance')  
        fun1()  
        print('after inhance')  
    return inner  
  
@decor  
def num():  
    print('hello mohit')  
  
num()
```

Output:-
befor inhance
hello mohit
after inhance

```
def num_decor(num):  
    def inner():  
        a = num()  
        add = a + 5  
        return add  
    return inner
```

```
@num_decor  
def num():  
    return 10
```

```
print(num())
```

Output:- 15

Example:-

```
def num_decor(num):  
    def inner():  
        a = num()  
        add = a + 5  
        return add  
    return inner
```

```
def num():  
    return 10
```

```
result = num_decor(num)  
print(result())
```

Output:- 15