Advanced Python

Subject: Decorators in Python

Lecturer : Reza Akbari Movahed

Decorators in Python

What is Decorators?

- Decorators are a very powerful and useful tool in Python since it allows programmers to modify the behaviour of a function or class.
- Decorators allow us to wrap another function in order to extend the behaviour of the wrapped function, without permanently modifying it.
- But before diving deep into decorators let us understand some concepts that will come in handy in learning the decorators.

Decorators in Python

First Class Objects

In Python, functions are first class objects which means that functions in Python can be used or passed as arguments.

Properties of first class functions:

- A function is an instance of the Object type.
- You can store the function in a variable.
- You can pass the function as a parameter to another function.
- You can return the function from a function.
- You can store them in data structures such as hash tables, lists, …

Decorators in Python

Decorators Definition

- As stated before, the decorators are used to modify the behaviour of function or class.
- In Decorators, functions are taken as the argument into another function and then called inside the wrapper function.

```
#Suppose we have defined a decorator called gfg_decorator

def hello_decorator():
    print("Gfg")

hello_decorator = gfg_decorator(hello_decorator)
```

Decorators in Python

```
def hello_decorator(func):

    def inner1():
        print("Hello, this is before function execution")

        func()

        print("This is after function execution")
    return inner1


def function_to_be_used():
    print("This is inside the function!!")


function_to_be_used = hello_decorator(function_to_be_used)

function_to_be_used()
```

step 2 · step 3 · step 4 · step 1 · step 5

```
def hello_decorator(func):

    def inner1():
        print("Hello, this is before function execution")

        func()

        print("This is after function execution")
    return inner1


def function_to_be_used():
    print("This is inside the function!!")


function_to_be_used = hello_decorator(function_to_be_used)

function_to_be_used()
```

step 6 · step 7 · step 8 · step 11 · step 9 · step 10 · step 12