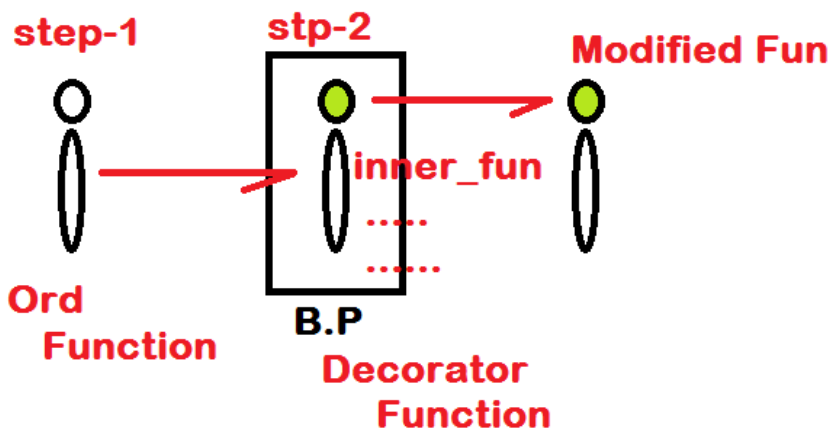
Decorators

- Decorator is a function which is used to provide some additional functionality for the function which is already existed

- step-1: Define ord function
-
- step-2: Define Dec_Function and pass an ordfunction as an argument to the dec_function
-
- step-3: Define Inner Function for making modification on the ord_function
-
- step 4: OuterFunction [Dec_function] need to return Inner Function [Where we made the modification]



Shas Kumar- SSSIT

Example 1:

```
def greet(name):
    print("Hello "+name+" Have A Nice Day..")

def dec_greet(func):   #func -> greet
    def inner(name):
        if name=="Sudha":
            print("Hello "+name+" Have  A Good Day ..")
        else:
            func(name)
    return inner  #returning hcode of inner

#calling
greet("Ramya")
greet("Sudha")

hcif=dec_greet(greet) #here we are passing hcode of greet
#print("hcif : ",hcif)  # hcif -- inner
hcif("Ramya")  # inner("Ramya")
hcif("Sudha")
```

Shashi Kumar - SSSIT

Example : 2

```
def division(x,y):
    z=x/y
    print("Result is : ",z)

def smart_division(func):
    def mydivision(x,y):
        if y==0:
            print("Values are not divided by Zero...")
        else:
            func(x,y)
    return mydivision

#calling
print("calling ord_function ")
division(10,3)
division(10,2)
#division(10,0) ZeroDivisionError

print("calling smart_division ")
myd=smart_division(division)   #myd is copy mydivision
myd(10,3)  # calling mydivision(10,3)
myd(10,2)
myd(10,0)
```

Shashi Kumar - SSSIT

Example 4:

```
def smart_division(func):
   def mydivision(x,y):
      if y==0:
         print("Values are not divided by Zero...")
      else:
         func(x,y)
   return mydivision

@smart_division
         # i.e division=smart_division(division)
def division(x,y):
   z=x/y
   print("Result is : ",z)

#calling
division(10,3)
division(10,0)
```

Decorators are classified into 2 types

> predefined decorators
>
> @staticmethod
>
> @classmethod

Shashi Kumar - SSSIT

@abstractmethod
@property ...

> userdefined decorators
@smart_division
@dec_greet

Example :

```
def dec_ssq(func):
   def wrapper(x):
      s=func(x)
      t=s*s
      return t
   return wrapper

@dec_ssq
def sq(x):
   s=x*x
   return s

#calling
r=sq(3)
print("Result is :",r)
```

Shashi Kumar - SSSIT