Department of Computer Science and Engineering
PES University, Bangalore, India

Lecture Notes
Python for Computational Problem Solving
UE23CS151A

Lecture #68

Problem solving using Closures and Decorators

By,
Prof. Sindhu R Pai,
Anchor, PCPS - 2023
Assistant Professor
Dept. of CSE, PESU

Verified by,
Prof. Sowmya Shree P,
Prof. Priyanka S
Assistant Professor,
Dept. of CSE, PESU

Many Thanks to
Dr. Shylaja S S (Director, CCBD and CDSAML Research Centers, Former Chairperson, CSE, PES University)
Prof. Chitra G M(Asst. Prof, Dept. of CSE, PCPS Anchor – 2022)

Problems on Closures and Decorators:

Solutions are available in this link:
https://drive.google.com/file/d/1IhjBrfd84teu3mfgCOfcYx86r4CvaJSG/view?usp=drive_link

1. Implement power function using closure i,e 3 raised to 4.
2. Implement an adder using closure.
3. Generate odd numbers using closure.
4. Implement a decorator program that defines a function called outer_div() that takes a function as input and checks the two numbers passed to function are following this condition – x >=y . If not swap x and y. Define a function inner() that takes two numbers as input and returns the result of dividing the larger number by the smaller number.
5. Take two integers from the user. Define a function - add_numbers() to add two numbers passed in the argument. Take two strings from the user. Define a function – concatenate_strings() to concatenate strings passed in the argument.

   Add a decorator function which takes type(example: str) as its argument and converts the return type of function to this specified type.

   Expected output: The type passed to decorator is str and hence both the functions returns the str type.

   Enter the number2
   Enter another number3
   Result: 5 <class 'str'>
   Enter first wordsindhu
   Enter second wordpai
   Result: sindhupai <class 'str'>

6. Define a function calculate_cube() to find the cube of the given number passed in the argument. Implement a decorator to validate function arguments based on a given condition. The condition must be passed in the argument of the decorator function.

   print(calculate_cube(5)) # Output: 125
   print(calculate_cube(-2)) #None
   Hint: use lambda

   Condition can be, whether the given number is positive or not.

- END -