Task-7: Implement Python generator and decorations. Aim: write a Python Program to implement python generator and decorator. 18 3 mont of 12 H ? ? Mail I'verite a Python Program that includes a generator function to Produce a Sequence of numbers when provided with stort b) Produce a default sequence of numbers stanking from o ending at 10, and with a step of if no values are provided Algorithm: end, and step values. Algorithm:

1. Define Generator Fonction

-> Define the function number sequence (start, end, step =) → Define the turier value?

3. Initialize current value?

> set coverent to value of stout. 3. Generate sequence:

-) while current is less than or equal to end.

-) yield the coverent value of coverent.

-) Increment current by stp. 3. Generate sequence: -> Read the starting number (start) from were reput-- Read the ending number (end) from user input -> Read the step value (step) from won in Put. -> create a generator object by calling number _ sequence 5. Create Generator obsect: (stoot, end, step) (stort, end, step)
6. Print Generated sequence:

Therate over values produced by generator object. -) Iterate over value for sit of modern levels and and levels of another levels of a Program: def number sequence (start, end, step=) current = start while convert <= end: Yield current coopent + = step start = int (input ("Enter the stanting number:")) end = int (input ('En les the ending number: ")) step = int (infint ("Enter the step value:"))

of create the generator sequence _ generator = number _ sequence requence - a generated sequence of normbers. for number in sequence-generatoris from print (number) of look until contact is los than n out hut:

The starting number: 1

Enter the ending number: 50

enter the step value: 5. output: of iterate over the generator object produced by onf. of 1 ter value in my generalists 6 11 ett Print each value Produced by generable 21 · (sulou) Init 26 31 36 41 7-1(b): Produce a default sequence of numbers starting from 46 7-1(b): Produce a derainer of if no values one Provided o, ending at 10, and with a step of 1 if no values one Provided Agorithm: Agosignin.

1. start Function:

Define the function my generator (n) that takes a parameter n

2. Initialize counter: - Set value to or rounded and prinom evolution and 3. Generate Values:

- while value is less than n: * Yield the current value: * Increment value by 1. 4. Create Generator object?

3 call my generator (11) to create a generator object.

5. Eterate and Print Values:

7 For each produced by generator object. Print value to octors of modern to constant the suffer contract of the start of the suffer of the su to lower ay 3. Define Greet goodfon:e took a oraclination shot is topped a function as tripped

or create the generator equence - garreso for = number - sequence Program: def my generator(n) & snows & beloveres soft thing def my - generator on set orange - somoupes ni redomin son # loop until counter is loss than n while value < n: totale value <n: The course of the counter yield value. # Increment the counter. value += 1

iterate over the generator object Produced by my generator for value in my_generator(3): # Print each value Produced by generator. Print (value). output: Imagine You are working on a messaging application that needs to format messages differently based on were preferences user can choose to have their menages automatically converted to upper car of to lower care. You are Provided with two decorators: upper con-decorator and lower con-decorator. there decorators modify the behavior of functions they decitate by conventing the text of the length of the select of the follow the state of the select of the sele by conventing the text to upper case or lower case, respectively. Algorithm: Algorithm:

1. Create Decorators;

Define uppercare decorator to convert the result of a function > Define lower cook - decorator to convert the result of a function to lower care:

2. Define functions:

-> Define Shout function to settorn input text. Apply this hundred 2. Define functions: -> Define whisper function to return the inPut fend. 3. Define Greet Function:

3. Define great function that: * Accepts a function as input.

prints the result works hours of 4. Execute the Program:

4. Execute the Print the greeting in upper care.

2. Coll greet to Print the greeting in lower care.

2. coll greet to Print the greeting. Program:

det upper cont decorator (func):

det wrapper (text): upper ()

return wrapper decorator (func):

det sower cont decorator (func):

det sower cont decorator (func): det wrappen (text): seturn func (text): lower() a upper cont decorator det shout (feet) is "txt-pol" tomon of the mayor return tent. d lower cox-decorator det sohisper (text): seturn text

seturn text

set great (func):

Jet great (func):

Jet great (func):

Jet greating = func ("Hi, I am created by a function payed as an angument.) S freeto dec. Progra m ! greet (shout) and band show of starting the great (whisher) I a mad have it tignalle of six how that a HI, IAM CREATED BY A FUNCTION PASSED AS AN ARGOMENT hi, iam created by a function passed of an argument. MANCE (5) RESULT AND ANALYSIS (5) VIVA VOCE (5) RECORD (5) TOTAL (20) 100 mg SIGN WITH DATE Result: Thus the Python Program to implement Python generator and decorators was successfully executed and the output was verified in so said month one diald consider object and also be used by a book object for enc ophion