

8.	17/09/25	U n it y w ork ing I m p l e m e n t P y t h o n G e n e r a t o r s A n d D e c o r a t o r s	15	15/09/25
----	----------	--	----	----------

1/9/25 Task-8: Implement Python Generator And Decorator

(a) Fibonacci Sequence Generator

Aim: To create a generator function that yields fibonacci numbers up to a given limit n and display the sequence.

Algorithm:

1. Define a generator function fibonacci-generator(n) that takes a maximum value.
2. Initialize the first two fibonacci numbers (0 and 1).
3. Yield the first number(0).
4. Use a while loop to generate subsequent fibonacci numbers until it exceeds the limit n.
5. Yield each fibonacci number until it exceeds the limit n.
6. Get user input for the maximum value.
7. Use the generator to iterate through and display the sequence.

Program:

```
def fibonacci_generator(n):
    """Generator function that yields fibonacci numbers upto n"""
    a, b = 0, 1
    yield a
    where b <= n:
        yield b
        a, b = b, a+b

def main():
    try:
        n = int(input("Enter the maximum value for fibonacci sequence:"))
        if n < 0:
            print("Please enter a non-negative number:")
            return
        print(f"fibonacci sequence upto {n}:")
        fib_gen = fibonacci_generator(n)
        for num in fib_gen:
            print(num, end=" ")
    print()
```

but not any more so that you can not see the end of the line

2024-03-29(1)

~~John Gossard~~ John Gossard

Output:-

Output:-

Enter the maximum value for fibonacci sequence = 50
Fibonacci sequence upto 50: 0 1 1 2 3 5 8 13 21 34

0 1 1 2 3 5 8 13 21 34

The author wants to add some figures to his paper
((Chlorophyll fluorescence))

(v) ~~adjective~~ without -ment = refers ~~to~~ ^{to} (adj) thing

```
except ValueError:  
    print("Please enter a valid integer")  
  
if __name__ == "__main__":  
    main()
```

Result:-

Thus, the program successfully creates a generator function that produces fibonacci numbers upto the specified limit.

11/25 (b) Function Execution Time Decorator

Aim: To implement a decorator that calculates and displays the execution time of any function, specifically applied to sorting function.

Algorithm:-

1. Create a decorator function timer-decorator that:
 - Records start time using time.time()
 - Calls the original function size to fit functionsize to fit
 - Records end time and calculate execution time.
 - Prints the execution time.
 - Returns the function result.
2. Create a function sort-random-list(size) that:
 - Generates a list of random numbers
 - Sorts the list using built-in sort.
 - Returns the sorted list
3. Apply the decorator to sorting functions
4. Test the different list sizes.

Program:-

```
import time
import random

def timer_decorator(func):
    def wrapper(*args, **kwargs):
        start_time = time.time()
        result = func(*args, **kwargs)
        end_time = time.time()
        execution_time = end_time - start_time
        print(f"function {func.__name__} executed in {execution_time} :")
        return result
    return wrapper

@timer_decorator
def sort_random_list(size):
    random_list = [random.randint(1, 1000) for _ in range(size)]
    sorted_list = sorted(random_list)
    return sorted_list

def main():
    sizes = [100, 500, 1000]
```

Output: Output matrix with size $m \times n$

Sorting list of size 1000; function 'sort-random-list' executed in 0.000998 seconds.

First 5 elements: [2, 4, 6, 8, 10]

Last 5 elements: 991, 992, 993, 995, 999

Sorting list of size 5000: To find a solution.

function 'sort-random-list' executed in 0.002995 seconds.
First 5 elements are [1, 1, 2, 2, 3]

First 5 elements: [1, 1, 2, 2, 3] Last 5 elements: [998, 998, 999, 999, 1000]

Last 5 elements: [998, 999, 1000, 1001, 1002]

(cont'd) following my visit
(approx. 1st exp't)

(Vaccines, Zephos) 3500000
G. Smith, Smith + Smith 200000

$$(\text{Liquor} \times 2 \text{ gml}^3) \text{ molar} = \text{Moles}$$

(3) $\text{soil} + \text{water} \rightarrow \text{soil} + \text{water}$

Final - Month - Solid state = solid polymer
ketones & - carboxylic acid by esterification

~~the following~~ without being
classified as ~~confidential~~

At $t = 0$, $(x(0), y(0)) = (0, 0)$. At $t = 1$, $(x(1), y(1)) = (0.818, 0.574)$.

~~(fusil Jackson) historie fra 1833-1862~~

```

for size in sizes:
    print(f"In sorting list of size {size}:")
    sorted_list = sort_random_list(size)
    print(f"first 5 elements: {sorted_list[:5]}")
    print(f"last 5 elements: {sorted_list[-5:]}")

if __name__ == "__main__":
    main()

```

~~but it is not working~~

VEL TECH - CSE	
EX NO.	8
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (20)	15
IN WITH DATE	20/9/22

Result:- Thus, the decorator successfully displays the execution time of the sorting function and measures and verify the execution time of the sorting function.

YB691NS