

Data Science & Artificial Intelligence



Python For Data Science

Lecture No.- 08



By- Satya sir

Recap of Previous Lecture



Recursion



Topics to be Covered



- OOPS Concepts
- default / Popular methods
- datetime module
- filter(), eval(), enumerator()

SUPER 1500+ - CLASS - 7 - Homework Question - 1

#Q. The return value would be _____

```
def f(L):
```

```
    i=1
```

```
    if len(L)==0:
```

```
        return 1
```

```
    else:
```

```
        return i+L[-1]
```

```
        f(L[:-1]) # Never Executes
```

```
List=[10,13,-12,34,67,15,22]
```

```
print(f(List[:-2]))
```

68

1+67

0	1	2	3	4	5	6
10	13	-12	34	67	15	22
-7	-6	-5	-4	-3	-2	-1

L:

0	1	2	3	4
10	13	-12	34	67
-5	-4	-3	-2	-1

L

0	1	2	3
10	13	-12	34
-4	-3	-2	-1

L

0	1	2
10	13	-12
-7	-6	-5

f(List[:-2])
 ↳ 1+67
 1+34
 1+(-12)
 1+13
 1+10

SUPER 1500+ - CLASS - 7 - Homework Question - 2

#Q. The output printed by below code is _____

```
def f(s):
```

```
    i=1
```

```
    if len(s)==0 :
```

```
        return
```

```
    else:
```

```
        print(s[len(s)-i], end=' ')
```

```
        f(s[2:])
```

```
string="GATE EXAM"
```

```
f(string)
```

A) M M M M

☒ B) M M M M M

C) M M M M M M M

D) M M M M M M M M M

GATE EXAM

$s[9-1] = s[8] = 'M'$

TE EXAM $s[7-1] = s[6] = 'M'$

- EXAM $s[5-1] = s[4] = 'M'$

XAM $s[3-1] = s[2] = 'M'$

M $s[1-1] = s[0] = 'M'$

[]

SUPER 1500+ - CLASS - 7 - Homework Question - 4

#Q. The output printed by below code is _____

```
def f(T):
    if len(T)==0 :
        return 0
    else:
        a,b,*c=T
        return a+b+f(T[2:])
t=(1,2,3,1,2,3,1,2,3,4)
print(f(t))
```

$$1+2+3+1+2+3+1+2+3+4$$

$$= 22$$

SUPER 1500+ - CLASS - 7 - Homework Question - 5

#Q. The output is _____

$$i = 6 + 0 = 6$$

$$6 + (-1)$$

$$= \underline{5}$$

```
def fun(s1,s2):
```

```
    if s1 is None or s2 is None :
```

```
        return -1
```

True

```
    else:
```

```
        i=len(s1.union(s2)) + len(s2.difference(s1))
```

```
        return i+fun((s1.intersection_update(s2)),s2.difference_update(s1))
```

$s1=\{1,2,3,4,2,5,1,2,6,4\} = \{1,2,3,4,5,6\} \Rightarrow \{1,2\}$

$s2=\{1,2,2,1,1,2,1\} = \{1,2\} \Rightarrow \{ \}$

```
print(fun(s1,s2))
```

A) 3

B) 4

☒ C) 5

D) 6

#Q. The output of below python code is _____

^{and} ^{False}
print(all([1, 2, 0, 4, -1, -5]), end=" ") ~~# True~~ ^{# False}
^{or} print(any([0, False, None, 0])) ~~# True~~ ^{# False}

- A) True True
- B) False True
- ☒ C) False False
- D) True False

#Q. The output of below python code is 568

```
print(0xA1 + 0o271 + 0b10111110 + 32)
```

Convert To Decimal

$$(A1)_H = A \times 16^1 + 1 \times 16^0 = 10 \times 16 + 1 = 161$$

$$(271)_8 = 2 \times 8^2 + 7 \times 8^1 + 1 \times 8^0 = 128 + 56 + 1 = 185$$

$$(10111110)_2 = 1 \times 2^7 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1$$

$$= 128 + 32 + 16 + 8 + 4 + 2$$

$$= 190$$

②

161

185

190

32

568

#Q. The output of below python code segment is _____

```
import datetime
```

```
x = datetime.datetime.now()
```

```
print(x.strftime("%a"), x.strftime("%B"), x.strftime("%y"))
```

Thu October 24

A) 10 Oct 2024

B) 10 October 2024

☒ C) Thu October 24

D) Thursday October 24

Directive	Description	Example
%a	Weekday, short version	Wed
%A	Weekday, full version	Wednesday
%w	Weekday as a number 0-6, 0 is Sunday	3
%d	Day of month 01-31	31
%b	Month name, short version	Dec
%B	Month name, full version	December
%m	Month as a number 01-12	12
%y	Year, short version, without century	18
%Y	Year, full version	2018
%H	Hour 00-23	17
%I	Hour 00-12	05
%p	AM/PM	PM
%M	Minute 00-59	41
%S	Second 00-59	08
%f	Microsecond 000000-999999	548513

#Q. The output of below python code segment is _____

```
import datetime
```

```
x = datetime.datetime.now()
```

```
print(x.strftime("%d"), x.strftime("%b"), x.strftime("%Y"))
```

Handwritten annotations above the code: "10" above "%d", "Oct" above "%b", and "2024" above "%Y".

- ✓ A) 10 Oct 2024
- B) 10 October 2024
- C) Thu October 24
- D) Thursday October 24



#Q. The final count value will be _____

X={12, 23, 25, 34, 45, 23, 25, 34, 45, 12}

Y={23, 25, 34, 12, 45, 56}

Z={45, 34, 12, 34, 45, 23, 56, 20}

count=1 \rightarrow False \Rightarrow True

if not(Y.isdisjoint(X)) and Y.issuperset(X):

count=count+1 # count=2

if X.isdisjoint(Z) or Z.issubset(Y):

count=count+2 X

elif X.issubset(Z) and Z.isdisjoint(Y):

count=count+3 X

else:

count=count+4 ✓ # count=6

print(count) # 6

isdisjoint \Rightarrow ! = No Common Elements

is subset \Rightarrow < =

is Superset \Rightarrow > =

X = {12, 23, 25, 34, 45}

Y = {23, 25, 34, 12, 45, 56}

Z = {45, 34, 12, 23, 56, 20}

#Q. The output will be _____

```
def myfunc(n):  
    return lambda a: a * n
```

```
mydoubler = myfunc(4)
```

```
print(mydoubler(10))
```

Handwritten annotations:
 - An arrow points from the `4` in `myfunc(4)` to the `a` in `a * n`.
 - The value `10` is written above the `a`.
 - The value `4` is written above the `n`.
 - The value `a=10` is written above the `10` in `mydoubler(10)`.
 - The value `40` is written below the `mydoubler(10)` expression.

A) 4

B) 8

C) 20

☒ D) 40

#Q. The output will be _____

List = [0, 1, 2, 3, 5, 9, 12]

```
result = filter(lambda x: x % 3 != x, List)
print(list(result))
```

- A) [1,2,3,5,9,12]
- B) [2,3,5,9,12]
- ☒ C) [3,5,9,12]
- D) [5,9,12]

$$0 \cdot 3 \neq 0 \text{ False}$$

$$1 \cdot 3 \neq 1 \text{ False}$$

$$2 \cdot 3 \neq 2 \text{ False}$$

$$3 \cdot 3 = 3 \text{ True}$$

$$5 \cdot 3 \neq 5 \text{ True}$$

$$9 \cdot 3 = 9 \text{ True}$$

$$12 \cdot 3 = 12 \text{ True}$$

#Q. What will be The Output if inputs are $x*(x+3)+(x-5)$ and 7 respectively?

```
def function():
    exp = input("Enter the function(in terms of x):")
    x = int(input("Enter the value of x:"))
    y = eval(exp)
    print("y =", y)
```

$$7*(7+3)+(7-5)$$

$$= 7*(10) + 2$$

$$= \underline{\underline{72}}$$

#Q. The output printed by below code is _____

```
List = ["abc", 5, "def", 7]
String = "GATE"
i = enumerate(List)
j = enumerate(String)
print (list(enumerate(List)))
print (list(enumerate(String)))
```

list of tuples

- A)
 - [(0, 'abc'), (1, 5, [2, 'def'], [3,7])]
 - [(2, 'G'), (3, 'A'), (4, 'T'), (5, 'E')]
- B)
 - [(0, 'abc'), (1, 5), (2, 'def'), (3,7)]
 - [(2, 'G'), [3, 'A'], [4, 'T'], [5, 'E']]
- C)
 - [(0, 'abc'), [1, 5], [2, 'def'], [3,7]]
 - [(2, 'G'), (3, 'A'), (4, 'T'), (5, 'E')]
- ☒ D)
 - [(0, 'abc'), (1, 5), (2, 'def'), (3,7)]
 - [(2, 'G'), (3, 'A'), (4, 'T'), (5, 'E')]

#Q. What will be the output of the following Python code?

class A:

```
def __init__(self):
    self.multiply(15)
    print(self.i)
```

A) 75

B) 60

C) 45

☒ D) 30

```
def multiply(self, i):
    self.i = 4 * i;
```

class B(A):

```
def __init__(self):
    super().__init__()
```

```
def multiply(self, i):
    self.i = 2 * i;
```

overwrites

= self.i = 30

obj = B()

#Q. The output printed will be _____

```
class A:
```

```
    def __init__(self):
```

```
        self.__i=3 # Private
```

```
        self.j=5 # Public
```

```
    def display(self):
```

```
        print(self.__i,self.j)
```

3, 6

```
class B(A):
```

```
    def __init__(self):
```

```
        super().__init__()
```

```
        self.__i=4
```

```
        self.j=6
```

```
c=B()
```

```
c.display()
```

Overwritten

A) 4,6

B) 3,5

C) 4,5

~~D) 3,6~~



2 mins Summary



$\circ \Rightarrow$ Public

$- \circ \Rightarrow$ Protected

$-- \circ \Rightarrow$ Private

- Miscellaneous Topic

A faint, dark background image showing a group of people, possibly a choir or a group of performers, in a dimly lit setting. The image is centered and serves as a backdrop for the text.

THANK - YOU