

▼ Day 4

Adhiraj Saha 2282013

1. Write a function `display_student()` that accepts student name, and her/his degree and display both. If the degree is missing in the function call, assign default value 'MCA' to it.

```
def display_student(student_name, degree="MCA"):
    print("Student Name:" , student_name , "and Degree:" , degree)

display_student("Adhiraj Saha")
display_student("Soumik Sil","MBA")
```

```
Student Name: Adhiraj Saha and Degree: MCA
Student Name: Soumik Sil and Degree: MBA
```

2. Write a function `attached()` that takes three parameters, the first a required parameter that is a number, the second a required parameter that is a string, and the third an optional parameter whose default is "". Returned value will be the first parameter, concatenated with the second, using the third.

```
def attached(x: int, y: str, z=""):
    return f"{x} {y} {z}"

a = 4
b = "Hi"
print(attached(a,b))
```

```
4 Hi
```

3. Define a function called `nums()` that has three parameters, the first, an integer, is required, the second parameter `mult_int` is optional with a default value of 10, the final parameter `switch`, is also optional with a default value of False. The function should multiply the two integers together, and if `switch` is True, should change the sign of the product before returning it.

```
def nums(f:int, mult_int = 10, switch=False):
    print(f*mult_int) if switch == True else print("ERROR")

nums(2,switch=True)
```

```
20
```

4. Write a function `add_mult()` that will accept two numbers and calculate the result of addition and multiplication of them and return both results in a single return statement.

```
def add_mult(x:int, y:int):
    return f"Addition and Multiplication of {x} and {y} is {x+y} and {x*y} respectively."
print(add_mult(3,6))
```

Addition and Multiplication of 3 and 6 is 9 and 18 respectively.

5. Write a function to return a tuple containing all the numbers in a list passed as an argument.

```
def listToTuple(li):
    t = tuple(li)
    print (t) if isinstance(t,tuple) else print("Not a tuple")
```

```
li=[]
x=0
n = int(input("Enter no of numbers: "))
for i in range(0, n):
    x += 1
    ele = int(input(f"Enter the number {x}: "))
    li.append(ele)
print(li) if isinstance(li,list) else print("Not a list")
listToTuple(li)
```

```
Enter no of numbers: 5
Enter the number 1: 4
Enter the number 2: 5
Enter the number 3: 2
Enter the number 4: 6
Enter the number 5: 1
[4, 5, 2, 6, 1]
(4, 5, 2, 6, 1)
```

6. $\sin(x)$ can be calculated approximately by summing the terms of the infinite series as follows

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} \dots$$

where x is expressed in radians. Write function(s) to calculate the value of $\sin x$, accepting x as degree from user. Compare your results with that of `math.sin()`.

```
import math
def pw(base, exponent):
    return base ** exponent
def fact(n):
    val = float(1)
```

```

    for i in range(1, n+1): val = val * i
    return val
def sin(x,n):
    val = x
    sign = -1
    i = 3
    while i < n: val = val + (pw(x, i)/fact(i) * sign); i = i + 2; sign = sign * -1
    return val
print(sin(0.7,200))
print(math.sin(0.7))

0.644217687237691
0.644217687237691

```

7. Write a function that accepts a list and returns a new list with unique elements of the first list.

```

def listToList(li):
    set1 = set(li)
    li2 = list(set1)
    return li2

li=[]
x = 0
n = int(input("Enter the number elements: "))
for i in range(n):
    x += 1
    ele = input(f"Enter the element {x}: ")
    li.append(ele)
print(listToList(li)) if isinstance(li,list) else print("Not a list")

Enter the number elements: 5
Enter the element 1: 3
Enter the element 2: 3
Enter the element 3: 4
Enter the element 4: 4
Enter the element 5: 1
['1', '3', '4']

```

8. Write a function that accepts a comma separated string of words as input and returns a colon- separated string of words after sorting them alphabetically.

```

def commaToColon(inp):
    li = inp.split(",")
    li.sort()
    print((":".join(li)) if isinstance(li, list) else print("Not a list"))

inp = str(input("Enter a comma separated string: "))
commaToColon(inp)

```

Enter a comma separated string: jfodfs,gfdgiosf,hgfhgfh,sdsdsad,fhgfh, fdsf,fds fdf
fdsf:fds fdf:fhgfh:gfdgiosf:hgfhgfh:jfodfs:sdsdsad

9. Write a function `last_char()` that takes a string as input, and returns only its last character. Use this function to sort list of strings by the last character of each string, from highest to lowest. Sample Input: ['150', '23', '781', '19', '3478', '12', '9005', '84', '9817', '96'] Output: ['19', '3478', '9817', '96', '9005', '84', '23', '12', '781', '150']

```
def last_char(n):
    return str(n)[-1]

n = ['150', '23', '781', '19', '3478', '12', '9005', '84', '9817', '96']
print(sorted(n,key=last_char))

['150', '781', '12', '23', '84', '9005', '96', '9817', '3478', '19']
```

10. Redo assignment 9. with lambda.

```
n = ['150', '23', '781', '19', '3478', '12', '9005', '84', '9817', '96']
sort = sorted(n,key=lambda n: str(n)[-1])
print(sort)

['150', '781', '12', '23', '84', '9005', '96', '9817', '3478', '19']
```

11. Sort a list of numbers based on their absolute values, writing your own function for calculating absolute value of a number and (i) using it in `sorted()` without lambda, (ii) using it in `sorted()` with lambda. Sample Input: [-5, -7, 4, -2, -9] Output: [-2, 4, -5, -7, -9]

```
#(i)
def abso(n):
    return ((n ** 2) ** 0.5 if isinstance(n,list) else "Not a list")
a = [-5, -7, 4, -2, -9]
print(sorted(a,key=abso))

#(ii)
b = [-5, -7, 4, -2, -9]
print(sorted(b,key=lambda n:(n ** 2) ** 0.5))

[-5, -7, 4, -2, -9]
[-2, 4, -5, -7, -9]
```

12. Sort a list of words first by their length, smallest to largest, and then alphabetically to break ties among words of the same length (using `sorted()` and lambda). Sample Input: ['mtech', 'btech', 'mca', 'bca', 'diploma', 'dsc'] Output: ['bca', 'dsc', 'mca', 'btech', 'mtech', 'diploma']

```
li = ['mtech', 'btech', 'mca', 'bca', 'diploma', 'dsc']
print(sorted(li, key=lambda x: (len(x), x)))

['bca', 'dsc', 'mca', 'btech', 'mtech', 'diploma']
```

13. Sort a list of words first by their length, largest to smallest, and then alphabetically to break ties among words of the same length (using sorted() and lambda). Sample Input: ['mtech', 'btech', 'mca', 'bca', 'diploma', 'dsc'] Output: ['diploma', 'btech', 'mtech', 'bca', 'dsc', 'mca']

```
li = ['mtech', 'btech', 'mca', 'bca', 'diploma', 'dsc']
print(sorted(li, key=lambda x: (-len(x), x)))

['diploma', 'btech', 'mtech', 'bca', 'dsc', 'mca']
```

14. Given a dictionary as stated in the sample input, use lambda to Sample Input: {"Kerala": ["Kannur", "Palakkad", "Thalassery"], "Maharastra": ["Bhandara", "Nagpur", "Wardha"], "West Bengal": ["Asansol", "Basirhat", "Tamluk"]} (i) sort the states in order by the first city name. (['West Bengal', 'Maharastra', 'Kerala']) (ii) sort the states by the length of the second city name, break ties of equal length by name of the second cities. (['Maharastra', 'West Bengal', 'Kerala']) (iii) sort the states in order by the number of cities having length greater than 6. (['Maharastra', 'Kerala', 'West Bengal'])

```
di = {"Kerala": ["Kannur", "Palakkad", "Thalassery"], "Maharastra": ["Bhandara", "Nagpur",
#(i)
sortedDict = dict(sorted(di.items(), key=lambda x: x[1]))
print(list(sortedDict.keys()))

#(ii)
print(sorted(di, key = lambda x: (len(di[x][1]), di[x][1])))

#(iii)
print(sorted(di, key=lambda x: sum([1 if len(i)>6 else 0 for i in di[x]])))

['West Bengal', 'Maharastra', 'Kerala']
['Maharastra', 'West Bengal', 'Kerala']
['Maharastra', 'Kerala', 'West Bengal']
```

15. Write a function that takes a string as a parameter and returns a list of the five most frequent characters in the string.

```
def freq(s:str):
    d={}
    for a in s:
```

```

        if a not in d.keys():
            d[a]=1
        else:
            d[a]+=1
    d_sorted=sorted(d.items(), key=lambda a:a[1], reverse=True)
    return d_sorted
li = freq("gjfoigjff dfdsofjofd,fd fdf fgeofkd f")
print("Five most frequent characters are:")
c=0
for a in li:
    print(a[0])
    c+=1
    if c==5:
        break

Five most frequent characters are:
f
d
o

g

```

16. Use lambda to sort a list of roll numbers by the last three digits of the roll number. Sample Input: [20223005, 20222342, 20229000, 20220002, 20222345, 20229329] Output: [20229000, 20220002, 20223005, 20229329, 20222342, 20222345]

```

li = [20223005, 20222342, 20229000, 20220002, 20222345, 20229329]
print(sorted(li, key = lambda x: x%1000))

[20229000, 20220002, 20223005, 20229329, 20222342, 20222345]

```

17. Use lambda to sort a list of names alphabetically by last name. Sample Input: ['Ales Bialiatski', 'Alain Aspect', 'Anton Zeilinger', 'Douglas Diamond'] Output: ['Alain Aspect', 'Ales Bialiatski', 'Douglas Diamond', 'Anton Zeilinger']

```

li = ['Ales Bialiatski', 'Alain Aspect', 'Anton Zeilinger', 'Douglas Diamond']
print(sorted(li, key=lambda x: x.split()[1]))

['Alain Aspect', 'Ales Bialiatski', 'Douglas Diamond', 'Anton Zeilinger']

```

18. Write a recursive function to get the list sum. [Test Data: [1, 2, [3,4], [5,6]] , Expected Result: 21]

```

li = [1, 2, [3,4], [5,6]]
print(li)
def listSum(li):
    if len(li) == 0: return 0
    if isinstance(li[0], list): return listSum(li[0]) + listSum(li[1:])

```

```

    return li[0] + listSum(li[1:])
print(f"Sum of the list: {listSum(li)}")

```

```

[1, 2, [3, 4], [5, 6]]
Sum of the list: 21

```

19. Write a recursive function to get the sum of digits of a non-negative integer.

```

def sumDigits(n):
    if n<10: return n
    return n//(10**(len(str(n))-1)) + sumDigits(n%(10**(len(str(n))-1)))
print(sumDigits(654))

```

```

15

```

20. Write a recursive function to find the greatest common divisor (gcd) of two integers.

```

def calGcd(x,y):
    if not x%y: return y
    return calGcd(y, x%y)
x = 65
y = 75
x,y=y,x if x<y else 0
print(calGcd(x,y))

```

```

5

```

21. Write a program to find number of characters, words, spaces and lines in a file.

```

with open("/content/drive/MyDrive/Classroom/[MCAP1102] Programming with Python [2022-2023]
    lns = file1.readlines()
print(f""""Characters: {sum([len(i) for i in lns])}
Words: {sum([(lambda x: len(x.split()))(i.strip()) for i in lns ]]}
Spaces: {sum([i.count(" ") for i in lns])}
Lines: {len(lns)}""")

```

```

Characters: 155
Words: 44
Spaces: 39
Lines: 7

```

22. Write a program to obtain the line number in which a given word is present.

```

with open("/content/drive/MyDrive/Classroom/[MCAP1102] Programming with Python [2022-2023]
    lns = file1.readlines()
w = "Ramesh"

```

```
l = 0
for i in lns:
    l+=1
    print(f"The word {w} is in line no {l}") if i.find(w) >= 0 else 0
```

The word Ramesh is in line no 3

23. Write a program to read the given (i) CSV file student_marks.csv, (ii) tab separated file student_marks.txt.

```
# (i)
file1 = open("/content/drive/MyDrive/Classroom/[MCAP1102] Programming with Python [2022-20
print(file1.read())
```

```
# (ii)
file2 = open("/content/drive/MyDrive/Classroom/[MCAP1102] Programming with Python [2022-20
print(file2.read())
```

```
,Gender,DOB,Maths,Physics,Chemistry,English,Biology,Economics,History,Civics
John,M,05/04/1988,55,45,56,87,21,52,89,65
Suresh,M,4/5/1987,75,55,,64,90,61,58,2
Ramesh,M,25/5/1989,25,54,89,76,95,87,56,74
Jessica,F,12/8/1990,78,55,86,63,54,89,75,45
Jennifer,F,2/9/1989,58,96,78,46,96,77,83,53
```

	Gender	DOB	Maths	Physics	Chemistry	English	Biology	Economics	History	Civics
John	M	05-04-88	55	45	56	87	21	52	89	65
Suresh	M	04-05-87	75	55		64	90	61	58	2
Ramesh	M	25-05-89	25	54	89	76	95	87	56	74

24. Write a program to accept string/sentences from the user till the user enters "END". Save the data in a text file and then display only those sentences which begin with 'T'.

```
li = []
print("Enter your strings/sentences and enter END to stop: ")
while True:
    text = input()
    if text.upper() == "END":
        break
    li.append(text)
file1 = open("/content/drive/MyDrive/Classroom/[MCAP1102] Programming with Python [2022-20
file1.writelines([i+"\n" for i in li])

file1 = open("/content/drive/MyDrive/Classroom/[MCAP1102] Programming with Python [2022-20
print(" ")
for ln in file1.readlines():
```



```
if ln[0] == "T":
    print(ln.strip())
```

```
Enter your strings/sentences and enter END to stop:
tnhdsd
Tjfidsjdso
fdosfksod
Tjfidsf
end
```

```
Tjfidsjdso
Tjfidsf
```

25. The sample file student.txt contains one line for each student. The student's name is the first thing on each line, followed by some exam scores. The number of scores might be different for each student. Using the text file student.txt write a program that prints out the names of students those have a total score more than 500.

```
file1 = open("/content/drive/MyDrive/Classroom/[MCAP1102] Programming with Python [2022-20
```

```
dict1={}
for i in file1:
    i = i.split()
    if not i: continue
    dict1[i[0]] = i[1:]
for key,val in dict1.items():
    score=0
    for i in val:
        score=score+int(i)
    if(score>500): print(f"{key} has a total score of {score}")
```

```
Ramesh has a total score of 556
Jessica has a total score of 545
Jennifer has a total score of 587
```

[Colab paid products](#) - [Cancel contracts here](#)

✓ 0s completed at 10:22 PM

