TASK ONE
NUMBERS AND VARIABLES

1. Create three variables in a single line and assign values to them in such a manner that each one of them belongs to a different data type.
E.g. :
(a,b,c) = (1,2.01,"string")

2. Create a variable of type complex and swap it with another variable of type integer.

```
(m,n) = (10+3j,1)
(m,n)=(1,10+3j)
print(m,n)
```

output:
1 (10+3j)

3. Swap two numbers using a third variable and do the same task without using any third variable.

```
x = 10
y = 5
z=x
x=y
y=z

print (x,y)
```

output:
5 10

```
(x,y)=(10,5)
(x,y)=(5,10)
print(x,y)
```

output:
5 10

4. Write a program that takes input from the user and prints it using both Python 2.x and Python 3.x Version.

```
a=int(input("enter 1 no:"))
```

```
b=int(input("enter 2 no:"))
sum = a+b
print(sum)
```

output:
enter 1 no:2
enter 2 no:3
5

5. Write a program to complete the task given below:
Ask users to enter any 2 numbers in between 1-10 , add the two numbers and keep the sum in another variable called z. Add 30 to z and store the output in variable result and print result as the final output.

```
x=5
y=6
z=x+y
u=30
z=z+u
print(z)
```

output:

41

6. Write a program to check the data type of the entered values.
HINT: Printed output should say - The data type of the input value is : int/float/string/etc

```
a=1.2
print(type(a))
```

output;
<class 'float'>

7. Create Variables using formats such as Upper CamelCase, Lower CamelCase, SnakeCase and UPPERCASE.
(Refer: https://capitalizemytitle.com/camel-case/)

Upper CamelCase: name = SnakeCaseName
Lower CamelCase: name = notCamelCase
SnakeCase: name = 'snake_case_name'
UPPERCASE: THE GIVEN STRING IN UPPERCASE = HELLO WORLD

8. If one data type value is assigned to ‘a’ variable and then a different data type value is assigned to ‘a’ again. Will it change the value? If Yes then Why?

It will change. It will replace old value with new one.

TASK TWO
OPERATORS AND DECISION MAKING STATEMENT

1. Write a program in Python to perform the following operation:
If a number is divisible by 3 it should print “Consultadd” as a string
If a number is divisible by 5 it should print “Python Training” as a string
If a number is divisible by both 3 and 5 it should print “Consultadd - Python Training” as a string.

```
a = int(input("Enter any value:"))
if a % 3 == 0 and a % 5 == 0:
    print("Consultadd Python Training")
elif a % 3 == 0:
    print("Consultadd")
elif a % 5 == 0:
    print("Python Training")
```

2. Write a program in Python to perform the following operator based task:
Ask user to choose the following option first:

If User Enter 1 - Addition
If User Enter 2 - Subtraction
If User Enter 3 - Division
If User Enter 4 - Multiplication
If User Enter 5 - Average
Ask user to enter two numbers and keep those numbers in variables num1 and num2 respectively for the first 4 options mentioned above.
Ask the user to enter two more numbers as first and second for calculating the average as soon as the user chooses an option 5.
At the end if the answer of any operation is Negative print a statement saying “NEGATIVE”
NOTE: At a time a user can only perform one action.

```
inp = eval(input("""Enter your choice
1. Addition
2. Subtraction
3. Division
4. Multiplication
5. Average
"""))
if inp in [1, 2, 3, 4, 5]:
    num1 = eval(input("Enter first number:"))
    num2 = eval(input("Enter second number:"))
    if inp == 1:
        print("Addition:", num1+ num2)
    elif inp == 2:
        print("Subtraction:", num1 - num2)
    elif inp == 3:
        print("Division:", num1/ num2)
    elif inp == 4:
        print("Multiplication:", num1* num2)
    elif inp == 5:
        first = eval(input("Enter the third number:"))
        second= eval(input("Enter fourth number:"))
        print("Average:", (num1+num2+first+second)/4)
else:
    print("Negative")
```

3. Write a program in Python to implement the given flowchart:

Program for the given flowchart

```
a, b, c = 10, 20, 30
avg = (a + b + c) / 3
print("avg = ", avg)
if avg > a and avg > b and avg > c:
    print("Average is higher than a, b, c")
else:
    if avg > a and avg > b:
        print("Average is higher than a, b, c")
```

```
    elif avg > a and avg > c:
        print("Average is higher than a, c")
    elif avg > b and avg > c:
        print("Average is higher than b, c")
    elif avg > a:
        print("Average is higher than a only.")
    elif avg > b:
        print("Average is higher than b only.")
    elif avg > c:
        print("Average is higher than c only.")
```

4. Write a program in Python to break and continue if the following cases occurs:
If user enters a negative number just break the loop and print “It’s Over”
If user enters a positive number just continue in the loop and print “Good Going”

```
while True:
    user = int(input("Enter a number: "))
    if user<0:
        print("it's over")
        break
    else:
        print("going good")
        continue
```

5. Write a program in Python which will find all such numbers which are divisible by 7 but are not a multiple of 5, between 2000 and 3200.

```
for i in range(2000,3201):
    if (i%7==0) and (i%5==0):
        print (i)
```

6. What is the output of the following code examples?

x=123

for i in x:
print(i)
i = 0

while i < 5:
print(i)
i += 1
if i == 3:
break
else:
print(“error”)
count = 0

while True:
print(count)
count += 1
if count >= 5:
Break

"""

```
x = 123
for i in x:
print(i)
```

```
i = 0
while i < 5:
    i += 1
    if i == 3:
        break
    else:
        print("error")

output:
error
error
```

```
count = 0
```

```
while True:
    print(count)
    count += 1
    if count >= 5:
        break

0
1
2
3
4
```

7. Write a program that prints all the numbers from 0 to 6 except 3 and 6.
Expected output: 0 1 2 4 5
Note: Use ‘continue’ statement

```
for i in range(7):
    if i==3 or i==6:
        continue
    print(i)
print('end')
```

8.
Sample input: consul72
Expected output: Letters 6 Digits 2

```
str = input("Input a string: ")
d=l=0
for c in str:
    if c.isdigit():
        d=d+1
    elif c.isalpha():
        l=l+1
    else:
        pass
print("Letters", l)
```

```
print("Digits", d)
```

9. Read the two parts of the question below:
Write a program such that it asks users to “guess the lucky number”. If the correct number is guessed the program stops, otherwise it continues forever.
Modify the program so that it asks users whether they want to guess again each time. Use two variables, ‘number’ for the number and ‘answer’ for the answer to the question whether they want to continue guessing. The program stops if the user guesses the correct number or answers “no”. (The program continues as long as a user has not answered “no” and has not guessed the correct number)

```
guess = 57
while True:
    user_input = int(input("Guess the lucky number: "))
    if user_input == guess:
        user_input, answer = input("Do you want to guess again? 
(Enter number & Yes to continue. Enter No to stop")
        if user_input != guess and answer.lower() == 'yes':
            continue
        elif user_input == guess:
            break
        elif answer.lower == 'no':
            break
```

10. Write a program that asks five times to guess the lucky number. Use a while loop and a counter, such as
counter=1

While counter <= 5:
print(“Type in the”, counter, “number”
counter=counter+1
The program asks for five guesses (no matter whether the correct number was guessed or not). If the correct number is guessed, the program outputs “Good guess!”, otherwise it outputs “Try again!”. After the fifth guess it stops and prints “Game over!”.

```
import random
guess = 57
counter = 0
```

```
while counter < 5:
    user_input = int(input("Guess the lucky number: "))
    if user_input == guess:
        print("Good guess!")
        guess = random.randint(1, 100)
    else:
        print("Try again!")
    counter += 1
print("Game over!")
```

11. In the previous question, insert break after the “Good guess!” print statement. break will terminate the while loop so that users do not have to continue guessing after they found the number. If the user does not guess the number at all, print “Sorry but that was not very successful”.

```
import random
guess = 57
counter = 0
while counter < 5:
    user_input = int(input("Guess the lucky number: "))
    if user_input == guess:
        print("Good guess!")
        break
    else:
        print("Try again!")
    counter += 1
print("Sorry but that was not very successful!")
```

TASK THREE
DATA STRUCTURES

1. Create a list of 10 elements of four different data types like int, string, complex and float.

```
a = [1,2.1,"anjali","byanjankar",5,6,7.9,8,9,10]
```

2. Create a list of size 5 and execute the slicing structure

```
l=[1,1.2,"Anjali",4,6]
print(l[0])
print(l[1])
print(l[2])
print(l[3])
print(l[4])
print(l[:])
```

3. Write a program to get the sum and multiply of all the items in a given list.

```
f = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
f1 = 0
f2 = 1
for i in f:
    f1 += i
for i in f:
    f2 *= i
print("Sum:", f1)
print("Multiply:", f2)
```

4. Find the largest and smallest number from a given list.

```
f = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
print("min:", min(f))
print("max:", max(f))
```

5. Create a new list which contains the specified numbers after removing the even numbers from a
predefined list.

```
f = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
l=[i for i in f if i%2!=0]
print(l)
```

6. Create a list of elements such that it contains the squares of the first and last 5 elements between
1 and30 (both included).

```
l= []
for i in range (1,31):
    l.append(i*i)
print(l[:5])
```

```
l= []
for i in range (1,31):
    l.append(i*i)
print(l[-5:])
```

```
f = [i*i for i in range(1, 31)]
print("First 5 elements:", f[:5])
print("Last 5 elements:", f[-5:])
```

7. Write a program to replace the last element in a list with another list.
Sample input: [1,3,5,7,9,10], [2,4,6,8]
Expected output: [1,3,5,7,9,2,4,6,8]

```
a = [1, 3, 5, 7, 9, 10]
b = [2, 4, 6, 8]

a[-1:]=b
print(a)
```

8. Create a new dictionary by concatenating the following two dictionaries:
Sample input: a={1:10,2:20} b={3:30,4:40}
Expected output: {1:10,2:20,3:30,4:40}

```
a={1:10,2:20}
b={3:30,4:40}
c={**a, **b}
print(c)
```

9. Create a dictionary that contain numbers in the form(x:x*x) where x takes all the values between 1
and n(both 1 and n included).
Sample input: n=5
Expected output: {1:1, 2:4, 3:9, 4:16, 5:25}

```
for x in range(1,n+1):
```

```
    d[x]=x*x

print(d)
```

10. Write a program which accepts a sequence of comma-separated numbers from console and
generates a list and a tuple which contains every number in the form of string.
Sample input: 34,67,55,33,12,98
Expected output: ['34','67','55','33','12','98'] ('34','67','55','33','12','98')

```
v= input("Input some comma seprated numbers : ")
list = v.split(",")
tuple = tuple(list)
print('List : ',list)
print('Tuple : ',tuple)
```

TASK FOUR
TRADITIONAL FUNCTIONS,ANONYMOUS FUNCTIONS &
HIGHER ORDER FUNCTIONS

1. Write a program to reverse a string.
Sample input: "1234abcd"
Expected output: "dcba4321"

Ans:

```
no= "1234abcd"
rev=no[::-1]
print(rev)
```

output:
dcba4321

2. Write a function that accepts a string and prints the number of uppercase letters and lowercase
letters.
Sample input: "abcSdefPghijQkl"
Expected Output: No. of Uppercase characters : 3 No. of Lower case Characters : 12

```
s="abcSdefPghijQkl"
d={"UPPER_CASE":0, "LOWER_CASE":0}
for c in s:
```

```
    if c.isupper():
        d["UPPER_CASE"]+=1
    elif c.islower():
        d["LOWER_CASE"]+=1
    else:
        pass
print ("No. of Upper case: ", d["UPPER_CASE"])
print ("No. of Lower case: ", d["LOWER_CASE"])
```

3. Create a function that takes a list and returns a new list with unique elements of the first list.

```
a = set([1,1,2,2,3,3,4,4])
m = list(a)
print(m)
```

4. Write a program that accepts a hyphen-separated sequence of words as input and prints the words
in a hyphen-separated sequence after sorting them alphabetically.

```
n = input ("Enter hyphen-separated sequence of words :- ")
a = n.split("-")
l.sort()
print('-'.join(l))
```

5. Write a program that accepts a sequence of lines as input and prints the lines after making all
characters in the sentence capitalized.
Sample input: Hello world Practice makes man perfect
Expected output: HELLO WORLD PRACTICE MAKES MAN PERFECT

```
a="Hello world Practice makes man perfect"
print(a.upper())
```

output:
HELLO WORLD PRACTICE MAKES MAN PERFECT

6. Define a function that can receive two integral numbers in string form and compute their sum and

print it in the console.

```
num1 = '10'
num2 = '20'
sum = (num1, num2)
print("sum is =", sum)
```

7. Define a function that can accept two strings as input and print the string with the maximum length
in the console. If two strings have the same length, then the function should print both the strings line
by line.

8. Define a function which can generate and print a tuple where the values are square of numbers
between 1 and 20 (both 1 and 20 included).

```
d=(i**2 for i in range(1,21))
print(d)
```

9. Write a function called showNumbers that takes a parameter called limit. It should print all the
numbers between 0 and limit with a label to identify the even and odd numbers.
Sample input: show Numbers(3) (where limit=3)
Expected output:
0 EVEN
1 ODD
2 EVEN
3 ODD

```
num = int(input("Enter a number: "))
mod = num % 2
if mod > 0:
    print("This is an odd number.")
else:
    print("This is an even number.")
```

10. Write a program which uses filter() to make a list whose elements are even numbers between 1
and 20 (both included)

```
l = []
```

```
for i in range(1,21):
    l.append(i**2)
print(l)
```

11. Write a program which uses map() and filter() to make a list whose elements are squares of even
numbers in [1,2,3,4,5,6,7,8,9,10].
Hints: Use filter() to filter even elements of the given listUse map() to generate a list of squares of the
numbers in the filtered list. Use lambda() to define anonymous functions.

```
a = [1,2,3,4,5,6,7,8,9,10]
w=list(map(lambda val:val*2, a))
print(w)
```

```
output;
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

12. Write a function to compute 5/0 and use try/except to catch the exceptions

```
try:
    numerator=int(input('enter the num:'))
    denominator=int(input('enter deno:'))
    result = numerator/denominator
    print(result)

except:
    print('code error')

print('pass')
```

output:
enter the num:5
enter deno:0
code error
pass

13. Flatten the list [1,2,3,4,5,6,7] into 1234567 using reduce().

```
from functools import reduce
i=reduce(lambda a,d: 10*a+d, [1,2,3,4,5,6,7], 0)
print(i)
```

14. Write a program in Python to find the values which are not divisible by 3 but are a multiple of 7.
Make sure to use only higher order functions.

```
def multiple(m, n):
    return True if m % n == 0 else False

print(multiple(3, 7))
```

15. Write a program in Python to multiply the elements of a list by itself using a traditional function
and pass the function to map() to complete the operation.

```
def multiplyList(m) :

    # Multiply elements one by one
    result = 1
    for x in m:
         result = result * x
    return result

# Driver code
list1 = [1, 2, 3]
list2 = [3, 2, 4]
print(multiplyList(list1))
print(multiplyList(list2))
```

16. What is the output of the following codes:
2
(i)
def foo():
try:
return 1

finally:
return 2
k = foo()
print(k)

```
def foo():
    try:
        return 1
    finally:
        return 2
k = foo()
print(k)

output:
2
```

(ii) def a():
try:
f(x, 4)
finally:
print('after f')
print('after f?')
a()