**##=> LIBRARIES:**

#2

1. cmath (sin,asin,cos…);
2. from numpy.lib.scimath import sqrt, sin;
3. from math import log10, exp, pi, cos, tan

#4

1. import pprint

**##=> SPECIAL FUNCTION OR ANYTHING ELSE:**

#2

1. print("""

...

...

...

... """) --> here the way you type exactly in that way the output will show!!

1. print('0' \* 5) --> output: 00000

#3

1. a.strip()
2. using + sign we can concatenate lists/tuples. Eg. [0,5]+[2,5] == [0,5,2,5]
3. test = [0]\*10 (Creates a list containing 10 zeros) only int numbers are allowed
4. a,b,c = 10,15,88 (assing each value to each variables)
5. Integer Division: (total // len(numbers))
6. Formatted Output: print("%7d\t%10d" %(tempC[i], tempF[i]))
7. A tuple with only one element requires a comma.

#4

1. enumerate() -> automatically generate index for items in for loop.
2. for updating any list element value you need to

access the element using index. Not by “for number in numbers” style.

1. The zip function turns n lists into one list of

tuples where each tuple contains n elements (known as an n-tuple).

1. @ temps = [[tC, tF] for tC, tF in zip(tempC, tempF)] (python table)
2. @ print(numbers[-1::-1]) ... reverse order

#5

1. To change the value of a global variable within a function the variable must be declared as global inside the function.(10th slide)
2. When a function is defined some of the parameters may be given default values, these parameters are known as keyword parameters. The other parameters are known as positional parameters. The keyword parameters must come after the positional parameters in the function header.
3. A Doc String is a string enclosed in triple quotes and may span several lines.
4. functionName.\_\_doc\_\_ (To print out the Doc String for a function use)
5. lambda parameters: expression (lambda function syntax)
6. value1 if condition else value2 (The inline if test syntax); 30th slide

#6

1. The function input is given a string (known as a prompt) to display and always returns a value of type 'str'.
2. We could use the eval function to covert the string returned by input to a numeric value.[if any string is input using eval ' ' quote is must]
3. eval(expression, globals=None, locals=None) (eval syntax)

**##=> LECTURES COVERINGS**

**#Week\_2:**

1. Real, Imaginary Numbers.
2. While Loop.
3. Quadratic Equation.
4. Summing Series.
5. Trajectory of Ball.
6. Relational and Boolean Operators.
7. Finding Leading Zeroes.
8. Computing Value of e

#Week\_3

1. List And Tuple
2. List -> append, insert, del, test.index(), test[-1]<- indicates last item of a lists
3. tuple -> index . Properties - Cant accidently change the content, execution is fast than list.
4. sum() function returns sum of list and tuples.
5. Reverse elements of List/test.reverse(), formatted output, end=''
6. A string can be thought of as a tuple of characters.

#Week\_4

1. range() function. As arguments, only Integers are allowed.
2. 11th, 36th slide.
3. List Comprehension.
4. Nested List.
5. Tables.
6. Extracting Sub-lists or Slices

#Week\_5

1. Function -> as list comprehension, value of function as parameter. Can return multiple values(tuple of values)
2. Local(high priority) And Global Variable
3. Doc string
4. Lambda Function
5. If Statement

#Week\_6

1. User Input
2. try-except
3. draw an X (pattern drawing)
4. pyplot