Level 1:

1. Given 2 strings, stra and strb, create a new string by appending strb in the middle of stra
2. Write a code to print the following pattern using a loop:

1

1. 2

1 2 3

1 2 3 4

1 2 3 4 5

1. Given a huge string, display only the first 25% (aprox) of character.

For example, if string is “greater good”, then the output should be “gre”.

1. Given a list of numbers, if the size of list is odd, then display the center most number. If the size of list is even then display average of the two numbers ate the center.
2. Take two lists and write a program that returns a list containing only the elements that are common between the two lists (without duplicates).
3. Write a code to check if a given year is leap year or not.
4. Given a string, count the number of upper case & lower-case characters. For example, given a string like “MassMutuAL”, the output should be: 4 upper & 6 lower
5. Write a code to check if the given number is prime or not.
6. Understand the usage of enumerate function. Write an example code to demonstrate its use.
7. Explore the use of **nonlocal** keyword. Write a code demonstrate difference between local, nonlocal and global data.

Level 2:

1. Iterate over a list and count the occurrence of each element. Create a dictionary to show the count of each element
2. Given a list of integers, remove all duplicates and put them in a tuple (only duplicates). Find the minimum, maximum, sum and average of all numbers in that list.
3. A bank has options for recurring deposit with an yearly interest rate of 8% cumulatively. Create a function which takes the deposit amount, and number of years (n) as an input. The function would return the amount cumulated at the end of ‘n’ years.
4. Write a code to extract all values from a multi-level nested dictionary. It should work for any level and should display only values and not keys.
5. Explore the super keyword for calling parent \_\_init\_\_() from a child class.
6. Given a list of strings, traverse through the list and if it comes across a string called as “reverse”, print the reverse of the list up to whatever has been traversed.

For example, input:

lista = [‘the’, ‘greater’, ‘good’, ‘is’, ‘reverse’, ‘of’, ‘what’, ‘exists’]

output:

[‘is’, ‘good’, ‘greater’, the’]

Level 3:

1. The time spent by an individual practicing Python

is 4 hours and 30 minutes on Saturday, 2 hours and 40 minutes on Sunday.

Can you create another object "Total" which will be an addition of the time spent on Saturday and Sunday.

Code to check if the method was implement correctly:

* sat = time(4, 30)
* sun = time(2, 40)
* total = sat + sun
* total.display()  
  Expected Output: 7 hours and 10 minutes

1. The class below holds marks of a student in various subjects.  
   This class is used to create objects which store marks of various students  
   Define another method which can be used to compare the average marks of two students

Code Skeleton

class Marks:

def \_\_init\_\_(self, C, java, python):

self.C = C

self.java = java

self.python = python

Input:

student\_1 marks [60, 70, 55] for C, java and python respectively

student\_2 marks [70, 50, 67] for C, java and python respectively

Calculation Logic for average grades:

if student\_1 > student\_2:

print("student one has a better average")

elif student\_1 < student\_2:

print("student two has a better average")

else:

print("both have the same average")

Expected Output: "student two has a better average"