

Module 1 - Getting Started with Python

Problem Statement:

The computations on several operations can be a tedious task for some. Your company has assigned a task for you to minimize the effort and enhance the productivity for several of these tasks. Simple tasks like finding the larger number among the lot, or changing the digits of a given number, etc.

Tasks To be Performed:

1. Create a python file named Module.
 - a. Inside the file, define 4 methods named – addition, subtraction, multiplication, and division.
 - b. Each method should only accept 2 arguments and should return the result of the operation performed in each method. For e.g., addition() should return the sum of two arguments.
 - c. Save the Module file in .py format.
2. Open a new python file and import the Module.py file.
 - a. Now call the 4 methods from the Module.py file, i.e., addition(), subtraction(), multiplication(), and division().
3. From the Module file, import only the addition() and pass the arguments so that it can display the result from the method.
4. From the Module file, import only the subtraction() and pass the arguments so that it can display the result from the method.
5. From the Module file, import both the multiplication() and division() and pass the arguments so that it can display the result from the methods.
6. Create a python if-else program to check if the given numbers are greater or not, also check whether the given number is an armstrong number or not, and check whether the given number is a prime number or not. Make use of python if-else, and elif statements for the same.
7. Create a calculator program for addition, subtraction, multiplication and floor division. Take the inputs from the user, and based on the choice of operation, return the results.
8. Create a fibonacci sequence using python if-else statements.
9. Create a nested dictionary with values as a nested list for each key in the dictionary.
10. Create two sets and perform the following:
 - a. Union of the two sets
 - b. Intersection of the two sets
11. Create a nested tuple from the dictionary, with each item in the tuple as a key value pair from the dictionary.
12. Create a list of the first 50 even natural numbers, and perform the following operations.

- c. Count the number of elements in the list.
 - d. Reverse the sequence of the list.
 - e. Sort the list in ascending and descending order.
 - f. Get the index value for the element 44, and update the element with the number 100.
 - g. Return a copy of the list, with the resulting list containing the square of each element.
13. Create a nested dictionary and perform the following operations.
- h. Return a list with the key value pairs from the dictionary.
 - i. Return a list with just the keys from the dictionary.
 - j. Remove the first and last key value from the dictionary.
 - k. Update the last key value pair in the dictionary after removing in the 3rd step.
14. For the given strings A = "Python Programming Language", B = "Best in the World" , perform the following operations.
- l. Using indexing operations, get the text "gram" from the string A.
 - m. Using indexing operations, get the text "World" from the string B.
 - n. Change the letters in both strings to Uppercase letters.
 - o. Concatenate the two strings.
15. Create a list with n numbers, and using negative indexing return the list starting from the 5th index to the n-2th index.
16. Using range(), create a list with numbers ranging from 5-100, and the number of elements should be exactly 19.