**Module – 3 (Collections, functions and Modules)**

**1) What is List? How will you reverse a list?**

A List in Python is an ordered, mutable (changeable) collection of items. Lists can contain elements of different data types (integers, strings, floats, etc.).

Lists are defined using square bracket [ ].

my\_list = [1, 2, 3, 4, 5]

my\_list.reverse()

print(my\_list)

**2) How will you remove last object from a list?**

my\_list = [1, 2, 3, 4, 5]

my\_list.pop()

print(my\_list)

**Suppose list1 is [2, 33, 222, 14, and 25], what is list1 [-1]?**

list1 = [2, 33, 222, 14, 25]

print(list1[-1]) # Output: 25

**3) Differentiate between append ( ) and extend ( ) methods?**

append() adds a single element to the end of the list.

extend() adds multiple elements to the list by extending it with another iterable.

**4) Write a Python function to get the largest number, smallest num and sum**

**of all from a list.**

def analyze\_list(numbers):

largest = max(numbers)

smallest = min(numbers)

total\_sum = sum(numbers)

return largest, smallest, total\_sum

my\_list = [10, 20, 4, 45, 99]

largest, smallest, total = analyze\_list(my\_list)

print("Largest:", largest)

print("Smallest:", smallest)

print("Sum:", total)

**5) How will you compare two lists?**

list1 = [1, 2, 3]

list2 = [1, 2, 3]

print(list1 == list2)

**6) Write a Python program to count the number of strings where the string length is 2 or more and the first and last character are same from a given list of strings.**

def match\_ends(words: List[str]) -> int:

count = 0

for word in words:

if len(word) >= 2 and word[0] == word[-1]:

count += 1

return count

**7) Write a Python program to remove duplicates from a list.**

def remove\_duplicates(input\_list: List) -> List:

unique\_list = []

seen = set()

for item in input\_list:

if item not in seen:

unique\_list.append(item)

seen.add(item)

return unique\_list

**8) Write a Python program to check a list is empty or not.**

user\_list = input("Enter elements of the list separated by spaces (leave blank for an empty list): ").split()

if not user\_list:

print("The list is empty.")

else:

print("The list is not empty.")

**9) Write a Python function that takes two lists and returns true if they have at least one common member.**

def have\_common\_member(list1, list2):

return bool(set(list1) & set(list2))

list1 = [1, 2, 3, 4]

list2 = [5, 6, 7, 3]

if have\_common\_member(list1, list2):

print("The two lists have at least one common member.")

else:

print("The two lists have no common members.")

**10) Write a Python program to generate and print a list of first and last 5 elements where the values are square of numbers between 1 and 30.**

def generate\_square\_list():

squares = [x\*\*2 for x in range(1, 31)]

result = squares[:5] + squares[-5:]

return result

square\_list = generate\_square\_list()

print("First and last 5 elements of the list:", square\_list)

**11) Write a Python function that takes a list and returns a new list with unique elements of the first list.**

def get\_unique\_elements(input\_list):

unique\_list = list(set(input\_list))

return unique\_list

original\_list = [1, 2, 2, 3, 4, 4, 5, 6, 6, 7]

unique\_list = get\_unique\_elements(original\_list)

print("Original List:", original\_list)

print("Unique List:", unique\_list)

**12) Write a Python program to convert a list of characters into a string.**

def list\_to\_string(char\_list):

return ''.join(char\_list)

char\_list = ['H', 'e', 'l', 'l', 'o', ' ', 'W', 'o', 'r', 'l', 'd']

result = list\_to\_string(char\_list)

print("List of characters:", char\_list)

print("Converted string:", result)

**13) Write a Python program to select an item randomly from a list.**

import random

def select\_random\_item(input\_list):

return random.choice(input\_list)

items = ['apple', 'banana', 'cherry', 'date', 'elderberry']

random\_item = select\_random\_item(items)

print("Original List:", items)

print("Randomly Selected Item:", random\_item)

**14) Write a Python program to find the second smallest number in a list.**

def find\_second\_smallest(numbers):

if len(numbers) < 2:

return "List must contain at least two distinct numbers."

unique\_numbers = sorted(set(numbers))

if len(unique\_numbers) < 2:

return "List must contain at least two distinct numbers."

return unique\_numbers[1]

numbers = [4, 2, 7, 1, 9, 1, 4]

result = find\_second\_smallest(numbers)

print("Original List:", numbers)

print("Second Smallest Number:", result)

**15) Write a Python program to get unique values from a list**

def get\_unique\_values(input\_list):

unique\_values = list(set(input\_list))

return unique\_values

numbers = [1, 2, 2, 3, 4, 4, 5, 6, 6, 7]

unique\_numbers = get\_unique\_values(numbers)

print("Original List:", numbers)

print("Unique Values:", unique\_numbers)

**16) Write a Python program to check whether a list contains a sub list**

def contains\_sublist(main\_list, sub\_list):

for i in range(len(main\_list) - len(sub\_list) + 1):

if main\_list[i:i+len(sub\_list)] == sub\_list:

return True

return False

main\_list = [1, 2, 3, 4, 5]

sub\_list = [3, 4]

if contains\_sublist(main\_list, sub\_list):

print("The list contains the sublist.")

else:

print("The list does not contain the sublist.")

**17) Write a Python program to split a list into different variables.**

my\_list = [1, 2, 3, 4]

a, b, c, d = my\_list

print("a:", a)

print("b:", b)

print("c:", c)

print("d:", d)

**18) What is tuple? Difference between list and tuple.**

A tuple is an immutable, ordered collection of elements. Like a list, it can store multiple items, but once a tuple is created, its elements cannot be modified, added, or removed.

The main difference between tuples and lists is that tuples are immutable, meaning their contents cannot be changed after creation, while lists are mutable and can be modified. Additionally, tuples are more memory-efficient compared to lists. Tuples can't be changed after they're created, but lists can be modified.

**19) Write a Python program to create a tuple with different data types.**

my\_tuple = (42, "Hello, World!", 3.14, True, [1, 2, 3], {"key": "value"})

print("Tuple with different data types:", my\_tuple)

for item in my\_tuple:

print(f"Element: {item}, Type: {type(item)}")

**20) Write a Python program to create a tuple with numbers.**

number\_tuple = (10, 20, 30, 40, 50)

print("Tuple with numbers:", number\_tuple)

for number in number\_tuple:

print("Element:", number)

**21) Write a Python program to convert a tuple to a string.**

def tuple\_to\_string(input\_tuple):

return ''.join(map(str, input\_tuple))

my\_tuple = ('H', 'e', 'l', 'l', 'o', ' ', 'W', 'o', 'r', 'l', 'd')

result = tuple\_to\_string(my\_tuple)

print("Original Tuple:", my\_tuple)

print("Converted String:", result)

**22) Write a Python program to check whether an element exists within a Tuple**

def check\_element\_in\_tuple(input\_tuple, element):

return element in input\_tuple

my\_tuple = (10, 20, 30, 40, 50)

element\_to\_check = 30

if check\_element\_in\_tuple(my\_tuple, element\_to\_check):

print(f"Element {element\_to\_check} exists in the tuple.")

else:

print(f"Element {element\_to\_check} does not exist in the tuple.")

**23) Write a Python program to find the length of a tuple.**

def get\_tuple\_length(input\_tuple):

return len(input\_tuple)

my\_tuple = (10, 20, 30, 40, 50)

length = get\_tuple\_length(my\_tuple)

print("The tuple is:", my\_tuple)

print("Length of the tuple:", length)

**24) Write a Python program to convert a list to a tuple.**

def list\_to\_tuple(input\_list):

return tuple(input\_list)

my\_list = [1, 2, 3, 4, 5]

converted\_tuple = list\_to\_tuple(my\_list)

print("Original List:", my\_list)

print("Converted Tuple:", converted\_tuple)

**25) Write a Python program to reverse a tuple.**

def reverse\_tuple(input\_tuple):

return input\_tuple[::-1]

my\_tuple = (1, 2, 3, 4, 5)

reversed\_tuple = reverse\_tuple(my\_tuple)

print("Original Tuple:", my\_tuple)

print("Reversed Tuple:", reversed\_tuple)

**26) Write a Python program to replace last value of tuples in a list.**

def replace\_last\_value(input\_list, new\_value):

return [t[:-1] + (new\_value,) for t in input\_list]

tuple\_list = [(1, 2, 3), (4, 5, 6), (7, 8, 9)]

new\_value = 100

updated\_list = replace\_last\_value(tuple\_list, new\_value)

print("Original List of Tuples:", tuple\_list)

print("Updated List of Tuples:", updated\_list)

**27) Write a Python program to find the repeated items of a tuple.**

def find\_repeated\_items(input\_tuple):

counts = {}

for item in input\_tuple:

counts[item] = counts.get(item, 0) + 1

repeated\_items = [key for key, value in counts.items() if value > 1]

return repeated\_items

my\_tuple = (1, 2, 3, 2, 4, 5, 3, 6, 7, 3)

repeated\_items = find\_repeated\_items(my\_tuple)

print("Original Tuple:", my\_tuple)

print("Repeated Items:", repeated\_items)

**28) Write a Python program to remove an empty tuple(s) from a list of tuples.**

def remove\_empty\_tuples(input\_list):

return [t for t in input\_list if t]

tuple\_list = [(1, 2), (), (3, 4), (), (5,)]

filtered\_list = remove\_empty\_tuples(tuple\_list)

print("Original List of Tuples:", tuple\_list)

print("Filtered List of Tuples:", filtered\_list)

**29) Write a Python program to unzip a list of tuples into individual lists.**

def unzip\_tuples(tuple\_list):

return list(map(list, zip(\*tuple\_list)))

tuple\_list = [(1, 2), (3, 4), (5, 6)]

unzipped = unzip\_tuples(tuple\_list)

print("Original List of Tuples:", tuple\_list)

print("Unzipped Lists:", unzipped)

**30) Write a Python program to convert a list of tuples into a dictionary.**

def tuples\_to\_dict(tuple\_list):

return dict(tuple\_list)

tuple\_list = [(1, 'apple'), (2, 'banana'), (3, 'cherry')]

converted\_dict = tuples\_to\_dict(tuple\_list)

print("Original List of Tuples:", tuple\_list)

print("Converted Dictionary:", converted\_dict)

**31) How will you create a dictionary using tuples in python?**

def tuples\_to\_dict(tuple\_list):

return dict(tuple\_list)

tuple\_list = [(1, 'apple'), (2, 'banana'), (3, 'cherry')]

converted\_dict = tuples\_to\_dict(tuple\_list)

print("Original List of Tuples:", tuple\_list)

print("Converted Dictionary:", converted\_dict)

**32) Write a Python script to sort (ascending and descending) a dictionary by value.**

d = {'a': 2, 'b': 1, 'c': 3}

ascending = dict(sorted(d.items(), key=lambda item: item[1]))

descending = dict(sorted(d.items(), key=lambda item: item[1], reverse=True))

print(ascending)

print(descending)

**33) Write a Python script to concatenate following dictionaries to create a new one.**

d1 = {'a': 1, 'b': 2}

d2 = {'c': 3, 'd': 4}

d3 = {\*\*d1, \*\*d2}

print(d3)

**34) Write a Python script to check if a given key already exists in a dictionary.**

d = {'a': 1, 'b': 2}

key = 'a'

print(key in d)

**35) How Do You Traverse Through A Dictionary Object In Python?**

d = {'a': 1, 'b': 2}

for k, v in d.items():

print(k, v)

**36) How Do You Check The Presence Of A Key In A Dictionary?**

d = {'a': 1, 'b': 2}

key = 'a'

print(key in d)Write a Python script to print a dictionary where the keys are numbers between 1 and 15.

d = {i: i for i in range(1, 16)}

print(d)

**37) Write a Python program to check multiple keys exist in a dictionary.**

d = {'a': 1, 'b': 2}

keys = ['a', 'b']

print(all(key in d for key in keys))

**38) Write a Python script to merge two Python dictionaries.**

d1 = {'a': 1, 'b': 2}

d2 = {'c': 3, 'd': 4}

d1.update(d2)

print(d1)

**39) Write a Python program to map two lists into a dictionary.**

keys = ['a', 'b']

values = [1, 2]

d = dict(zip(keys, values))

print(d)

**40) Write a Python program to combine two dictionary adding values forcommon keys.**

**d1 = {'a': 100, 'b': 200, 'c':300} o d2 = {'a': 300, 'b': 200,’d’:400}**

**Sample output: Counter ({'a': 400, 'b': 400,’d’: 400, 'c': 300}).**

from collections import Counter

d1 = {'a': 100, 'b': 200, 'c': 300}

d2 = {'a': 300, 'b': 200, 'd': 400}

result = Counter(d1) + Counter(d2)

print(dict(result))

**41) Write a Python program to print all unique values in a dictionary.**

d = {'a': 1, 'b': 2, 'c': 1, 'd': 3}

unique\_values = set(d.values())

print(unique\_values)

**42) Why Do You Use the Zip () Method in Python?**

list1 = [1, 2, 3]

list2 = ['a', 'b', 'c']

result = zip(list1, list2)

print(list(result))

**43) Write a Python program to create and display all combinations of letters, selecting each letter from a different key in a dictionary.**

**Sample data: {'1': ['a','b'], '2': ['c','d']}**

**Expected Output:**

**ac ad bc bd**

from itertools import product

data = {'1': ['a', 'b'], '2': ['c', 'd']}

combinations = [''.join(item) for item in product(\*data.values())]

print(combinations)

**44) Write a Python program to find the highest 3 values in a dictionary.**

d = {'a': 50, 'b': 100, 'c': 200, 'd': 300, 'e': 150}

highest\_values = sorted(d.values(), reverse=True)[:3]

print(highest\_values)

**45) Write a Python program to combine values in python list of dictionaries.**

**Sample data: [{'item': 'item1', 'amount': 400}, {'item':'item2', 'amount':300}, o {'item': 'item1', 'amount':750}]**

**Expected Output:**

**Counter ({'item1': 1150, 'item2': 300})**

from collections import Counter

data = [{'item': 'item1', 'amount': 400}, {'item': 'item2', 'amount': 300}, {'item': 'item1', 'amount': 750}]

result = Counter()

for entry in data:

result[entry['item']] += entry['amount']

print(result)

**46) Write a Python program to create a dictionary from a string.Note: Track the count of the letters from the string.**

**Sample string: 'w3resource'**

**Expected output:**

**{'3': 1,’s’: 1, 'r': 2, 'u': 1, 'w': 1, 'c': 1, 'e': 2, 'o': 1}**

from collections import Counter

string = 'w3resource'

result = Counter(string)

print(dict(result))

**47) Write a Python function to calculate the factorial of a number (a nonnegative integer)**

def factorial(n):

if n == 0 or n == 1:

return 1

else:

return n \* factorial(n - 1)

number = 5

print(f"The factorial of {number} is: {factorial(number)}")

**48) Write a Python function to check whether a number is in a given range.**

def is\_in\_range(n, start, end):

return start <= n <= end

usageprint(is\_in\_range(5, 1, 10))

**49) Write a Python function to check whether a number is perfect or not.**

def is\_perfect(n):

return n == sum(i for i in range(1, n) if n % i == 0)

print(is\_perfect(6))

**50) Write a Python function that checks whether a passed string is palindrome or not.**

def is\_palindrome(s):

return s == s[::-1]

print(is\_palindrome('madam'))

**51) How Many Basic Types Of Functions Are Available In Python?**

There are three types of functions in the python programming language. These are: Built-in functions- There are 68 built-in functions in the Python programming language. User-defined functions- To define a user-defined function, the def keyword is used.

**52) How can you pick a random item from a list or tuple?**

import random

items = [1, 2, 3, 4]

print(random.choice(items))

**53) How can you pick a random item from a range?**

import random

print(random.choice(range(1, 10)))

**54) How can you get a random number in Python?**

import random

print(random.random())

**55) How will you set the starting value in generating random numbers?**

import random

random.seed(10)

print(random.random())

**55) How will you randomize the items of a list in place?**

import random

items = [1, 2, 3, 4]

random.shuffle(items)

print(items)

**56) Write a Python program to read a random line from a file.**

import random

def read\_random\_line(file\_path):

with open(file\_path, 'r') as file:

lines = file.readlines()

return random.choice(lines).strip()

file\_path = 'sample.txt' # Replace with the path to your file

print(read\_random\_line(file\_path))

**57) Write a Python program to convert degree to radian.**

import math

def degree\_to\_radian(degree):

return degree \* (math.pi / 180)

degree = 90

radian = degree\_to\_radian(degree)

print(f"{degree} degrees is equal to {radian} radians")

**58) Write a Python program to calculate the area of a trapezoid.**

def trapezoid\_area(base1, base2, height):

return ((base1 + base2) / 2) \* height

base1 = 10

base2 = 15

height = 7

area = trapezoid\_area(base1, base2, height)

print(f"The area of the trapezoid is: {area}")

**59) Write a Python program to calculate the area of a parallelogram.**

def parallelogram\_area(base, height):

return base \* height

base = 10

height = 5

area = parallelogram\_area(base, height)

print(f"The area of the parallelogram is: {area}")

**60) Write a Python program to calculate surface volume and area of a cylinder**

import math

def cylinder\_volume(radius, height):

return math.pi \* radius\*\*2 \* height

def cylinder\_surface\_area(radius, height):

return 2 \* math.pi \* radius \* (radius + height)

radius = 5

height = 10

volume = cylinder\_volume(radius, height)

surface\_area = cylinder\_surface\_area(radius, height)

print(f"Volume of the cylinder: {volume}")

print(f"Surface area of the cylinder: {surface\_area}”)

**61) Write a Python program to returns sum of all divisors of a number.**

def sum\_of\_divisors(n):

return sum(i for i in range(1, n + 1) if n % i == 0)

number = 12

result = sum\_of\_divisors(number)

print(f"The sum of all divisors of {number} is: {result}”)

**62) Write a Python program to find the maximum and minimum numbers from the specified decimal numbers.**

def find\_max\_min(numbers):

return max(numbers), min(numbers)

decimal\_numbers = [3.14, 1.59, 2.65, 5.35, 4.87]

maximum, minimum = find\_max\_min(decimal\_numbers)

print(f"Maximum: {maximum}")

print(f"Minimum: {minimum}")