FUNCTION ASSIGNMENT

Name – Amandeep singh

Reg Email – amancooldeep94@gmail.com

Course Name – Data Analytics July'24

Assignment Name – Function Assignment

Submission Date – 10/08/2024

Git Link -

Q.1) What is the difference between a function and a method in Python?

Ans) Function: A function is a block of code that performs a specific task. It can be called independently without being tied to any object.

```
def Square(a):
    return (a ** 2)
Square(5)
```

```
25
```

Method: A method is a function that is associated with an object. It is called on an object and can access and modify the data contained in the object.

2. Explain the concept of function arguments and parameters in Python.

Ans) Parameters: Parameters are the variables listed inside the parentheses in the function definition.
Arguments: Arguments are the values that are passed to the function when it is called.

```
def add(a, b):                # a and b are parameters
return a + b
result = add (3, 5)           # 3 and 5 are arguments
print(result)
```

3. What are the different ways to define and call a function in Python?

Ans) Defining a Function: Using the def keyword.
Calling a Function: Using the function name followed by parentheses.

```
# Defining a function
```

```
def say_hello():
return "Hello, World!"
# Calling a function
print(say_hello())
```

```
Hello, World!
```

4. What is the purpose of the `return` statement in a Python function?

Ans) The return statement is used to exit a function and go back to the place where it was called. It can also send back a value to the caller.

```
def square(n):
    return n ** 2
result = square(4)
print(result)
```

```
16
```

5. What are iterators in Python and how do they differ from iterables?

Ans) Iterable: An object that can return an iterator (e.g., lists, tuples, strings).

Iterator: An object that represents a stream of data; it returns data one element at a time when next() is called.

```
# Iterable
my_list = [1, 2, 3]
# Iterator
my_iter = iter(my_list)
print(next(my_iter))
```

```
1
```

```
print(next(my_iter))
```

```
2
```

```
print(next(my_iter))
```

```
3
```

6. Explain the concept of generators in Python and how they are defined.

Ans) Generators are a special class of iterators that allow you to iterate through a sequence of values. They are defined using the yield keyword.

```
def countdown(n):
```

```
    while n > 0:
        yield n
        n -= 1

for number in countdown(5):
    print(number)
```

```
5
4
3
2
1
```

7. What are the advantages of using generators over regular functions?

Ans) Memory Efficiency: Generators do not store all the values in memory, they generate values on the fly.

Lazy Evaluation: Values are produced only when needed, which can improve performance for large datasets.

8. What is a lambda function in Python and when is it typically used?

Ans) A lambda function is an anonymous function defined with the lambda keyword. It is used for creating small, one-off functions without a name.

```
# Lambda function
add_lambda = lambda a, b: a + b
print(add_lambda(3, 5))
```

```
8
```

9. Explain the purpose and usage of the `map()` function in Python.

Ans) The map() function applies a given function to each item of an iterable and returns a map object (an iterator).

```
def square(x):
    return x * x
numbers = [1, 2, 3, 4]
squared_numbers = map(square, numbers)
print(list(squared_numbers))
```

```
[1, 4, 9, 16]
```

10. What is the difference between `map()`, `reduce()`, and `filter()` functions in Python?

Ans) map(): Applies a function to all items in an iterable.

```
numbers = [1, 2, 3, 4]

squared = map(lambda x: x * x, numbers)

print(list(squared))
```

```
[1, 4, 9, 16]
```

reduce(): Applies a function cumulatively to the items of an iterable, reducing the iterable to a single value.

```
from functools import reduce
total = reduce(lambda x, y: x + y, numbers)
print(total)
```

```
10
```

filter(): Filters items out of an iterable, returning only those that match a condition.

```
even_numbers = filter(lambda x: x % 2 == 0, numbers)

print(list(even_numbers))
```

```
[2, 4]
```

11. Using pen & Paper write the internal mechanism for sum operation using reduce function on this given list:[47,11,42,13];