Assigenment -03

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

   => function is a built in types which is predefined in the python library for a specific kind of work. Eg print() is predefined function which is used to print in python.

   Method is manual function is which is not predefined ,but it is defined by the developers in the real time to perform the specific work. It is defined manually as demand of work

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

   => Parameters are the placeholders listed inside the parentheses when you defined the function , they act as the var to receive the values

   Eg greet(name):

   print(f“hello {name}”) here ‘name ‘ is the parameter passed in to the function defined

   Arguments:- It is the actual values passed in to the function when it is called

   Eg ;- greet(‘afaque’) here ‘afaque’ is the args passed into the function when it is called.

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

   => There are many ways to define and call a function

   Function with no parameters and no return types

   ```
   def welcome():
     print("Welcome to the Student Marks System!")
   ```

Calling function welcome()

Functions with parameters and returns

```
def calculate_average(name, *marks):
    average = sum(marks) / len(marks)
    return f"{name}'s average score is {average:.2f}"
```

Calling it

```
print(calculate_average("Afaque", 85, 90, 78, 92))
```

Function with defaults parameters

```
def disp_student(name='unknown',course='cse'):
        print(f"name{name} course {course}")
```

Calling the function

```
disp_student('afaque')
disp_student()
```

using **kwargs in function

```
def student_info(**details):
    for key, value in details.items():
        print(f"{key}: {value}")

student_info(name="Afaque", age=22, roll=101)
```

Lambda Function

```
grade=lambda score:"pass" if score>=40 else "fail"
print(grade(75))
```

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

=> Whenever a function executes a return statement , it ends the function and gives back a value wherever the function was called

5. What are iterators in Python and how do they differ from iterables?
=> An iterable is an any object in python which is capable of returning of an element ,when passed to an iter() function to get an iterator
Eg list, set, tuple.Dict, we can loop over them using for loop

=> iterators is an object that represent stream of data it is returned by calling iter() on iterable

6. Explain the concept of generators in Python and how they are defined.
=> Generators in python is a special type of iterators, that are used.generate values at a time as they are needed , rather then story the values in memory all at once. Eg def and yield instead of return

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

=> Memory Efficiency
Generators produce items one at a time using lazy evaluation, meaning they don't store the entire sequence in the memory

Faster Startup:- since generator yield values as needed , they start producing output immediately without processing the entire sequence at a time

8. . What is a lambda function in Python and when is it typically used?
=>Lambda is the small anonymous function defined using the keyword lambda .
=> we can any number of args but only one expression is allowed
=>returns the value of expression automatically .
=> Lambda function is commonly used with sorted(), filter(),and map()

9. Explain the purpose and usage of the `map()` function in Python.
=>The map() function in python is used to apply a function to each item in iterable(like list ,tuple) and return new map object which is iterator .

Purpose:- to transform data efficiently without writing an explicit loop.

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

map(function,iterable)
purpose:- Applies a function to **each item** in an iterable., A new iterable with transformed items.

Filter(function,iterable)
Purpose:-filters items in an iterable based on condition.
Return :- A new iterable with only items where the function returns `True`.

reduce(function,iterable)
Purpose:-repetedly applies a function to the accumulated result and the next item