Experiment No. :- 09

① what are positional (required) arguments?

Ans:-
Positional arguments are passed to a function in the same order as parameters defined in the function.

Ex:-
```
def add (a, b):
        return a+b
    x=add (5, 10)
    print ("sum =", x)
```

② write a program using positional arguments.

Ans:-
```
def product (a, b):
      print ("product =", a*b)
product (4, 6)
```

③ what are keyword arguments?

Ans:-
Keyword arguments are passed using parameters names. order doesn't matter.

Ex:-
```
def display (name, age)
          print (name, age)
    display (age = 20, name = "Ravi")
```

④ write a program using keyword arguments

Ans:-
```
def student (name, course):
          print ("Name:", name)
          print ("course :", course)
student (course = "Python", name = "Anu")
```

output:-

```
sum =15
```

output:-

```
Product = 24
```

output:-

```
Ravi 20
```

output:-

```
Name: Anu
Course: Python
```

⑤ what are default arguments?

Ans:-
Default arguments have predefined values. If no value is passed, the default value is used.

EX:-
```
def greet (name="student"):
        print ("Hello ", name)
greet()
```

⑥ write a program using default arguments

Ans:-
```
def bill (amount=100):
        print ("Bill amount :", amount)
bill (500)
bill ()
```

⑦ what are arbitary arguments?

Ans:- Arbitary arguments are used when the number of arguments is not fixed. They uses *args and **kwargs.

⑧ write. a program using *args..

Ans:-
```
def total (*nums):
        print ("sum = ", sum (nums))
total (10,20,30)
```

output:-

```
Hello student
```

output:-

```
Bill amount : 500
Bill amount : 100
```

output:-

```
sum = 60
```

⑨ write a program using **kwargs .

Ans:-

```
def employee (**details):
    for k, v in details.items():
        print (k, ":", v)
employee (name="kiran", Id=101, dept="IT")
```

⑩ Differentiate between *args and **kwargs.

Ans:-

*args stores values as tuple.

**kwargs stores values as dictionary.

⑪ what is lambda function?

Ans:- A lambda function is a small anonymous function without any name and its main purpose is one time usage or instant.

Syntax:- Lambda input arguments : Expression.

EX:-

```
s=lambda n : n*n
print (s(4))
```

⑫ what is map function?

Ans:- A map function applies a function to each element of an iterable i.e. list, tuple, etc. A new sequence is created from the existing data. From each element of sequence a new sequence is generated.

Syntax:- map (function, sequence)

EX:-

```
l= [1,2,3,4,5]
l1 = list (map (lambda n : n*n, l))
print(l1)
```

output :-

```
name : kiran
id : 101
dept : IT
```

output :-

```
16
```

output :-

```
[1,4,9,16,25]
```

⑬ what is Filter function?

Ans:- A filter function selects elements from an iterable based on a condition. It filters the values from the given sequence based on some condition. Syntax:- filter (function, sequence)

EX:-

```
nums = [1,2,3,4,5,6,7,8,9,10]
even-nums = list (filter (lambda x : x%2 ==0, nums))
print (even-nums)
```

⑭ what is reduced function?

Ans:- A reduce() function cumulatively to elements of an iterable and returns a single value.

Syntax:- reduce (function, sequence)

EX:-

```
from functools import reduce
n = [1,2,3,4,5]
total = reduce (lambda x,y : x+y, n)
print (total)
```

output :-

```
[2,4,6,8,10]
```

output :-

```
15
```