Python functions

# 1. Function syntax and creation

```
def message():
        print("Welcome to functions in Python")
message()
```

# 2. Creating a function with parameters

```
def course_func(name, course_name):
        print("Hello", name, "Welcome to MGIT")
        print("Your course name is", course_name)
### call function
course_func('John', 'Python')
```

#3. Creating a function with parameters and return value

```
def calculator(a, b):
   add = a + b
   # return the addition
   return add
# call function
# take return value in variable
res = calculator(20, 5)
print("Addition :", res)
```

4. Write a python program using function to return even numbers and odd numbers from a given list

```
def even_odd(list1):
    even_num = []
    odd_num = []
    for n in list1:
        if n % 2 == 0:
            even_num.append(n)
        else:
            odd_num.append(n)
    # return a list
    return even_num, odd_num

# Pass list to the function
evennums, oddnums = even_odd([2, 3, 42, 51, 62, 70, 5, 9])
print("Even numbers are:", evennums)
print("Odd numbers are:", oddnums)
```

5. Write a python program to demonstrate returning multiple values from a function

```
# Return multiple values
def arithmetic(num1, num2):
    add = num1 + num2
    sub = num1 - num2
    multiply = num1 * num2
    division = num1 / num2
    # return four values
    return add, sub, multiply, division
```

```
# read four return values in four variables
addition, subtraction, multiplication, division = arithmetic(10, 2)

print("Addition: ", addition)
print("Subtraction: ", subtraction)
print("Multiplication: ", multiplication)
print("Division: ", division)
```

6. Write a python program to demonstrate the following
i. Default arguments
ii. Variable-length arguments
iii. Recursive function

Default arguments take the default value during the function call. If we do not pass them. We can assign a default value to an argument in function definition using the = assignment operator.

```
# function with default argument
def message(name="Guest"):
   print("Hello", name)
# calling function with argument
message("John")
# calling function without argument
message()
```

#Variable-length Arguments

In Python, sometimes, there is a situation where we need to pass multiple numbers of arguments to the function. Such types of arguments are called variable-length arguments. We

can declare a variable-length argument with the * (asterisk) symbol.

```
def addition(*numbers):
   total = 0
   for no in numbers:
      total = total + no
   print("Sum is:", total)
# 0 arguments
addition()
# 5 arguments
addition(10, 5, 2, 5, 4)
# 3 arguments
addition(78, 7, 2.5)
```

#Recursive Function

A recursive function is a function that calls itself, again and again.

```
def factorial(no):
   if no == 0:
      return 1
   else:
      return no * factorial(no - 1)
checkfact = int(input("Enter a number:"))
print("factorial of a number is:", factorial(checkfact))
```