ANSWERS ASSIGNMENT – 5

TOPICS – FUNCTONS IN PYTHON

1) A function that accepts two values and find their sum.
Solution:

```
def sum(val1, val2):
    sum = val1 + val2
    print("sum = {}".format(sum))

sum(10, 20)
```

2) A python program to find the sum of two numbers and return the result from the function.
Solution:

```
def sum(val1, val2):
    sum = val1 + val2
    return sum

sum = sum(10, 20)
print(sum)
```

3) A function to test whether a number is even or odd.
Solution:

```
def evenOdd(num):
    if num % 2 == 0:
        print("{} is even".format(num))
    else:
        print("{} is odd".format(num))


evenOdd(4)
evenOdd(7)
```

4) A python program to test whether a number is even or odd.
Solution:

```
def evenOdd(num):
    if num % 2 == 0:
        print("{} is even".format(num))
    else:
        print("{} is odd".format(num))

evenOdd(4)
evenOdd(7)
```

5) A python program to calculate factorial values of numbers.
Solution:

```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)

n = int(input("Input a number to compute the factiorial : "))
print(factorial(n))
```

6) A python program to check if a given number is prime or not.
Solution:

```
num = 11

if num > 1:
    for i in range(2, num // 2):

        if (num % i) == 0:

            print(num, "is not a prime number")
            break
    else:
        print(num, "is a prime number")

else:
    print(num, "is not a prime number")
```

7) A python program that generates prime numbers with the help of a function to test prime or not.
Solution:

```
def prime(num):
    if num > 1:
        for i in range(2, num // 2):

            if (num % i) == 0:
                print(num, "is not a prime number")
                break
        else:
            print(num, "is a prime number")

    else:
        print(num, "is not a prime number")

userInput = int(input("Enter a any integer number : "))
prime(userInput)
```

8) A python program to understand how a function returns two values.
Solution:

```
def test2():
    return 'abc', 100, [0, 1, 2]
a, b, c = test2()

print(a)
print(b)
print(c)
```

9) A function that returns the result of addition, subtraction, multiplication and division.
Solution:

```
def add(a, b):
    return  "sum =", a + b

def subtraction(a, b):
    return  "subtraction =", a - b

def mul(a, b):
    return  "multiplication = ",a * b

def division(a, b):
    return  "division =", a / b
```

```
a, b = 20, 10
print(add(a, b))
print(subtraction(a, b))
print(mul(a, b))
print(division(a, b))
```

10) A python program to see how to assign a function to a variable.
Solution:

```
def sum(val1, val2):
    sum = val1 + val2
    return sum

# assign a function to a variable
addition = sum
print(addition(10, 20))
```

11) A python program to know how to define a functions inside another function.
Solution:

```
def outer(num1):
    def inner_increment(num1):
        return num1 + 1
    num2 = inner_increment(num1)
    print(num1, num2)

outer(21)
```

12) A python program to know how to pass a function as parameter to another function.
Solution:

```
def shout(text):
    return text.upper()

def whisper(text):
  return text.lower()

def greet(func):
    # storing the function in a variable
    greeting = func("Good Morning!")
    return greeting
```

```
upper = greet(shout)
print(upper)

lower = greet(whisper)
print(lower)
```

13) A python program to know how to pass a function as parameter to another function.
Solution:

```
def Square(X):
    return (X * X)

def SumofSquares(Array, n):
    Sum = 0
    for i in range(n):
        SquaredValue = Square(Array[i])
        Sum += SquaredValue
    return Sum

Array = [1, 2, 3, 4, 5]
n = len(Array)

Total = SumofSquares(Array, n)
print("Sum of the Square of List of Numbers:", Total)
```

14) A python program to pass an integer to a function and modify it.
Solution:

```
def modify(x):
    print("before modify x = ", x)
    x = 45
    return x

x = 10
y = modify(x)

print("after modify x = ", y)
```

15) A python program to pass a list to a function and modify it.
Solution:

```
def modify(list):
    print("before modify list = ", list)
    list = [47, 11]
    return list

lst = [1,2,3,4]

lst2 = modify(lst)
print("after modify list = ", lst2)
```

16) A python program to create a new object inside the function does not modify outside object.
Solution:

```
def outer(num1):
    #create new object inside function
    def inner_increment(num1):
        return num1 + 1

    num2 = inner_increment(num1)
    print(num1, num2)

outer(21)
```

17) A python program to understand the positional arguments of a function.
Solution:

```
def greet(*names):
   """This function greets all
   the person in the names tuple."""

   # names is a tuple with arguments
   for name in names:
       print("Hello",name)

greet("Monica","Luke","Steve","John")
```

18) A python program to understand the keyword arguments of a function.
Solution:

```
def greet(msg , name):
    print("msg =",msg, "\nname = ", name)
```

```
greet(name = "Bruce",msg = "How do you do?")
greet(msg = "How do you do?",name = "Bruce")
greet("How do you do?", name = "Bruce")
```

19) A python program to understand the use of default arguments in a function.
Solution:

```
def greet(name, msg = "Good morning!"):

   print("Hello",name + ', ' + msg)

greet("Kate")
greet("Bruce","How do you do?")
```

20) A python program to show variable length argument and its use.
Solution:

```
        #*args: Receive multiple arguments as a tuple
        #**kwargs: Receive multiple keyword arguments as a dictionary

def my_sum(*args):
    return sum(args)

print(my_sum(1, 2, 3, 4))
print(my_sum(1, 2, 3, 4, 5, 6, 7, 8))
```

21) A python program to understand keyword variable arguments.
Solution:

```
def myFun(**kwargs):
    for key, value in kwargs.items():
        print(key, " : ",  value)

myFun(first='Geeks', mid='for', last='Geeks')
```

22) A python program to understand global and local variables.
Solution:

```
# This function uses global variable s
```

```
def f():
    print("global var inside function :",g)

    l = "local variable"
    print(l)

g = "global variable"
f()
print("global var outside function",g)
```

23) A python program to access global variable inside a function and modify it.
Solution:

```
g = "global variable"
print("before modify : ", g)

def f():
    global g
    g = "global var modified"

f()
print("after modified  : ",g)
```

24) A python program to get copy of global variable into a function and work with it.
Solution:

```
def f():
    s = "Me too."
    print("copy of global variable", s)

# Global scope
s = "I love Geeksforgeeks"
f()
print("original global variable", s)
```

25) A function to accept a group of numbers and find their total average.
Solution:

```
num = int(input('How many numbers: '))
total_sum = 0
```

```
for n in range(num):
    numbers = float(input('Enter number : '))
    total_sum += numbers

avg = total_sum/num
print('Average of ', num, ' numbers is :', avg)
```

26) A function to display a group of strings.
Solution:

```
groupOfString = ["red","yellow","orange","green"]

def display(array):
    for i in array:
        print(i)

display(groupOfString)
```

27) A python program to calculate factorial values using recursion.
Solution:

```
def recur_factorial(n):
   if n == 1:
       return n
   else:
       return n*recur_factorial(n-1)
# take input from the user
num = int(input("Enter a number: "))
# check is the number is negative
if num < 0:
   print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
   print("The factorial of 0 is 1")


else:
   print("The factorial of",num,"is",recur_factorial(num))
```

28) A python program to solve Towers of Hanoi problem.
Solution:

29) A python program to create a lambda function that returns a square value of a given number.
Solution:

```
g = lambda x: x * x
print(g(7))
```

30) A lambda function to calculate sum of two numbers.
Solution:

```
s = lambda a, b : a + b
print(s(10, 20))
```

31) A lambda function to find the bigger number in two given numbers.
Solution:

```
b = lambda a, b : a if(a > b) else b
print(b(10,20))
```

32) A python program using filter() to filter out even numbers from a list.
Solution:

```
# Initialisation of list
list1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

is_even = lambda x: x % 2 == 0

# using filter


lis2 = list(filter(is_even, list1))

# Printing output
print(lis2)
```

33) A lambda that returns even numbers from a list.
Solution:

```
my_list = [3,5,2,11,6,8]
list_Even_Numbers = list(filter(lambda varX: varX % 2 == 0,my_list))
print("Following are Even numbers in the list =", list_Even_Numbers)
```

34) A python program to find squares of elements in a list.

Solution:

```
list1 = [1,2,3,4,5,6]
list2 = []

def squares(list):
    for i in list:
        list2.append(i * i)

squares(list1)
print(list2)
```

35) A lambda function that returns squares of elements in a list.

Solution:

```
list1 = [1,2,3,4,5]
arr2 = list(map(lambda x: x ** 2, list1))
print(arr2)
```

36).A python program to find the product of elements of a two different lists using lambda function.

Solution:

37).A lambda function to calculate products of elements of a list.

Solution:

```
from functools import reduce

product = reduce(lambda x, y: x*y, [1,2,3,4,5])
print(product)
```

38).A decorator to increase the value of a function by 2.

Solution:

39).A python program to apply a decorator to a function using @ symbol.

Solution:

40).A python program to create two decorators.

Solution:

41).A python program to apply two decorators to the same function using @ symbol.

Solution:

42).A python program to create a generator that returns a sequence of numbers from x to y.

Solution:

43) A generator that returns characters from A to C.

Solution:

44).A python program to calculate the gross salary and net salary of an employee.

Solution:

45).A python program that uses the functions of employee module and calculate the net and gross salaries of an employee.

Solution:

46).A python program using special __name__ variable.

Solution:

47).A python program that imports the previous Python program as a module.

Solution: