CHAPTER-03

WORKING WITH FUNCTIONS

1. What is the default return value for a function that does not return any value explicitly?

(a) None (b) int (c) double (d) Null

2. Which of the following items are present in the function header?

(a) Function name only (b) Both function name and parameter list

(c) Parameter list only (d) Return value

3. Which of the following keywords marks the beginning of the function block?

(a) Def (b) Define (c) def (d) Function

1. A_____is a subprogram that acts on data and often returns a value.
2. Python names the top-level segment (main program) as____.
3. In Python, program execution begins with the first statement of___segment.
4. The first line of function definition that begins with keyword def and ends with a colon (:)is also known as function call. (true/false)
5. Variables that are listed with the parentheses of function header are called function variables. (true/false)
6. In Python, the program execution begins with the first statement of the main segment. (true/false)
7. A python function may return multiple values. (true/false)
8. A void function also returns a value i.e., None to its caller. (true/false)
9. The values being passed through a function call statement are called_.
10. The values received in the function definition/header are called__.
11. What is the name given to that area of memory, where thesystem stores the parameters and local variables of a function call?

(a) A heap (b) Storage area (c) A stack (d) An array

12. Which of the following function headers is correct?

(a) def f(a = 1, b):

(b) def f(a = 1, b, c = 2):

(c) def f(a=1, b=1,c=2):

(d) def f(a=1, b=1, c=2, d) :

13. Which of the following statements is not true for parameter passing to functions?

(a) You can pass positional arguments in any order.

(b) You can pass keyword arguments in any order.

(c) You can call a function with positional and keyword arguments.

(d) Positional arguments must be before keyword arguments in a function call.

14. Which of the following function calls can be used to invoke the below function definition?

def test (a, b, c, d)

(a) test (1, 2, 3, 4)

(b) test (a=1, 2, 3, 4)

(c) test (a = 1, b = 2, c = 3, 4)

(d) test (a=1, b=2, c=3, d=4)

15. Which of the given argument types can be skipped from a function call?

(a) Positional arguments

(b) Keyword arguments

(c) Named arguments

(d) Default argument

16. A parameter having default value in the function header is known as a_.

17. A_____argument can be skipped in the function call statement.

18. _____arguments are the named arguments with assigned values being passed in thefunction call statement.

19. A void function also returns a_____value to its caller.

20. By default, Python names the segment with top-level statements (main program) as _____.

21. The_____refers to the order in which statements are executed during a program run.

22. The default value for a parameter is defined in function____.

True/ False Questions: -

23. Non-default arguments can be placed before or after a default argument in a functiondefinition.

24. A parameter having default value in the function header is known as a default parameter.

25. Default parameters cannot be skipped in function call.

26. The default values for parameters are considered only if no value is provided for the parameter in the function call statement.

APPLICATION BASED QUESTIONS

27. Write the output of the given code.

```
def addEm(x,y,z):
        Print(x+y+z)
def prod(x,y,z):
        return x*y*z
a=addEm(6,16,26)
b=prod(2,3,6)
Print(a,b)
```

28. Trace the flow of execution of the following code and also write its output.

```
1. def switch (x, y) :
2.       x, y = y,x
3.       print ("inside switch :", end = '  ')
4.       print ("x=", x, "y=",y)
5. x = 5
6. y = 7
7. print("x=", x, "y=", y)
```

8. switch(x,y)
9. print("x=",x,"y=", y)

29. Trace the flow of execution of the following code.

```
1.  def switch (x,y) :
2.      x, y = y,x
3.      print ("inside switch :", end = '  ')
4.      print ("x=", x, "y=",y)
5.      return
6.  x = 5
7.  y = 7
8.  print("x=", x, "y=", y)
8.  switch(x,y)
9.  print("x=",x,"y=", y)
```

30. Trace the flow of execution of the following code.

```
1.   #temperature conversion program
2.  def c_to_f(c) :
3.      result = c/5.0 * 9 + 32
4.      return result
5.
6.  tempf = c_to_f(19)
7.  print(tempf)
```

31. Trace the flow of execution of the following code.

```
1.  def c_to_f(c):
2.      result = c/5.0 * 9 + 32
3.      return result
4.
5.  tempf = c_to_f(19)
6.  print(tempf)
7.  c_to_f(19)
8.  print(“conversion over”)
```

32. Trace the flow of execution of the following code.

```
1.  def sum(a,b,c,d) :
2.      result = 0
3.      result=result + a + b + c + d
4.      return result
6.  def length():
7.      return 4
8.
9.  def mean(a,b,c,d) :
10.     return float(sum (a,b,c,d) ) /length( ))
12.   print (sum(a,b,c,d), length( ) , mean (a,b,c,d) )
```

Also find the output of the above code if a,b,c,d are 10,20,30,40 respectively.

33. Trace the flow of execution of the following code.

```
1.  def power (b, p):
2.      r = b ** p
3.      return r
4.
5.  def calcSquare(a):
6.      a = power (a, 2)
7.      return a
8.
9.  n = 5
10. result = calcSquare(n)
11. print(result)
```

Also find the output of the above code if a,b,c,d are 10,20,30,40 respectively.

34. Trace the flow of execution of the following code.

```
1.  def increase(x):
2.      x = x +1
3.
4.  # main program
5.  x = 3
6.  print(x)
7.  increase(x)
8.  print(x)
```

35. Trace the flow of execution of the following code.

```
1.  def increment(x):
2.      z = 45
3.      x = x + 1
4.      return x
5.
6.  # main
7.  y = 3
8.  print(y)
9.  y = increment(y)
10. print(y)
11. q = 77
12. print(q)
13. increment (q)
14. print(q)
15. print (x)
16. print (z)
```

36. Identify which of the following are legal/illegal function call statements with reason for the function header.

def interest (prin, cc, time = 2, rate = 0.09) :

(i) interest (5000, 3, rate = 0.05)
(ii) interest (rate = 0.05, 5000, 3)
(iii) interest (500, time = 2, rate = 0.05)
(iv) interest (prin = 3000, cc = 5)

37. Predict the output of the following code fragment?

```
def func(message, num = 1):
    print(message*num)
func('Python')
func('Easy',3)
```

38. Predict the output of the following code fragment?

```
def check (n1 = 1, n2 = 2):
    n1 = n1+n2
    n2 +=1
    print (n1, n2)
```

```
check( )
check(2, 1 )
check(3)
```

39. What is the Output of the following code?

```
a= 1
def f ( ) :
        a = 10
print(a)
```

40. What will be the output of the following code?

```
def interest (prnc, time = 2, rate = 0.10) :
        return (prnc*time*rate)
print(interest (6100,1))
print(interest (5000, rate = 0.05))
print(interest(5000,3,0,12))
print(interest(time = 4, prnc = 5000))
```

41. Consider a function with following header:

def info (object, spacing = 10, collapse = 1):

Here are some function calls given below. Find out which of these are correct and which of these incorrect stating reasons.

a. **info(obj1)**

b. **info(spacing=20)**

c. **info(obj2, 12)**

d. **info(obj11, object = obj12)**

e. **info(obj3, collapse = 0)**

f. **info( )**

g. **info (collapse = 0, obj3)**

h. **info(spacing = 15, object = obj4)**

42. Consider below given function headers. Identify which of these will cause error and why?

(i) def func (a = 1, b) :

(ii)def func (a = 1, b, c = 2):
(iii)def func (a = 1, b = 1, c = 2) :
(iv)def func (a = 1, b = 1, c =2, d) :

43. Following code intends to add a given value to global variable a. What will the following code produce ?

```
1. def increase(x) :
2. a = a + x
3. return4.
5.  a = 20
6.  b = 5
7.  increase(b)
8.  print(a)
```

44. From the program given below, identify the scope of each variable in the program.

```
X = 10

pi = 3.14

def incr(x) :

    y = x + 1

    return y
def area(r) :
    n = incr(r)
    ar = pi * n * n
    return ar
incr(5)
print(x)
area(4)
```

45. What are the errors in the following code? Correct the code and predict output :

```
total = 0;

def sum(arg1, arg2):

    total = arg1 + arg2;

    print("Total :",total)
```

```
        return total;
    sum(10,20);
    print("Total  :",total)
```

46. What will be the output of the following programs?

(i)

```
num = 1
def myfunc( ) :
    return num
print(num)
print(myfunc())
print(num)
```

(ii)

```
num = 1
def myfunc( )
    num = -10
    return num
print(num)
print(myfunc())
print(num)
```

(iii)

```
num = 1
def myfunc( ) :
    global num
    num = 10
    return num
print(num)
print(myfunc())
print(num)
```

(iv)

```
def display( ) ;
    print("Hello",end =')

display( )
print("there !")
```

47. Predict the output of the following code:

```
a = 10
y = 5
def myfunc( ) :
    y = a
    a=2
    print("y=",y,"a=",a)
    print ("a+y=",a+y)
    return a + y
print("y=", y, "a=",a)
print(myfunc( ) )
print("y=",y, "a=",a)
```