Visit QB365-SLMS - https://www.qb365.in/slms/ Complete Questions and Answers

Answers Key:

Long Answer Type Questions

1. def add(a, b):
   print("ADDING % d + % d" % (a, b))
   return a + b
   def subtract(a, b):
   print("SUBTRACTING % d - % d" % (a,b))
   return a - b
   def multiply(a, b):
   print ("MULTIPLYING % d*% d" %(a, b))
   return a * b
   def divide(a, b):
   print ("DIVIDING % d / % d" % (a, b))
   return a/b
   Now call these functions:
   >>> age = add (32, 25)
   #ADDING 32 + 25
   >>> age
   5 7
   >>> height = subtract (175, 165)
   #SUBTRACTING 175-165
   >>> height
   10
   >>> weight = multiply (75, 70)
   #MULTIPLYING 75 * 70
   >>> weight
   5250
   >>> IQ = divide (100, 50)
   #DIVIDING 100/50
   >>> IQ
   2
2. The list of identifiers used in a function call is called actual parameter(s) whereas the list of parameters used in the function definition is called formal parameter(s).
   Actual parameter may be value / variable or expression. Formal parameter is an identifier.
   Example
   def area(side): # line 1
   return side*side
   print(area(5)) # line2
   In line 1, side is the formal parameter and in line 2, while invoking area() function, the value 5 is the actual parameter.
   A formal parameter, i.e. a parameter, is in the function definition. An actual parameter, i.e. an argument, is in a function call.
3. The differences between a local variable and a global variable are given below:

Local Variable	Global Variable
These are declared within a function	These are declared outside all functions
Local variable is accessible only with in a function in which it is declared	Global variable is accessible throughout the program.
It is also called an internal variable	It is also called an external variable
The local variable of two functions may be declared with same name.	No two global variables in a program may have same name

   e.g. a = 10 #global variable
   def value (i) :
   c=0 #local variable
   c = a + i
   return c
   Now call the function value
   >>> value (5)
   Output
   15
4. In Python, global keyword allows the programmer to modify the variable outside the current scope.
   It is used to create a global variable and make changes to the variable in local context. A variable declared inside a function is by default local and a variable declared outside the function is global by default. The keyword global is written inside the function to use its global value. Outside the function, global keyword has no effect

Example
c=10 # global variable
def add( ):
global c
c = c + 2 # global value of c is incremented by 2
print("Inside add( ):", c)
add()
c=15
print("In main:",c)
Output
Inside add( ): 12
In main: 15

5. In Python, global keyword allows the programmer to modify the variable outside the current scope.
It is used to create a global variable and make changes to the variable in local context. A variable declared inside a function is by default local and a variable declared outside the function is global by default. The keyword global is written inside the function to use its global value. Outside the function, global keyword has no effect
Example
c=10 # global variable
def add( ):
global c
c = c + 2 # global value of c is incremented by 2
print("Inside add( ):", c)
add( )
c=15
print("In main:",c)
Output
Inside add( ): 12
In main: 15

Short Answer Type Questions

6. def CountVowels (s) :
c = 0
for ch in s :
if ch in 'aeiouAEIOU' :
c += 1
return c

QB 365 Study Material
QUESTION BANK 365

7. A parameter having default value in the function header is known as a default parameter. A parameter can be specified as default only if all the parameters on its right are specified as default. To specify a default parameter syntax is
def<function header> (arg1, arg2---arg x =<value>);
argx will now be a default parameter with default value as the value specified.

8. • In Python, variables declared inside a function are local by default.
• If we want to modify a global variable from inside a function, we must declare it with the global keyword.
• This tells Python that we are referring to the variable in the global scope, not creating a new local variable.
• Without global, changes inside the function affect only a local copy, leaving the global variable unchanged.

9. A global variable is a variable that is accessible globally i.e everywhere the program. A local variable is one that is only accessible to the current scope, such as temporary variables used in a single function definition.
A variable declared outside of the function or in global scope is known as global variable. This means, global variable can be accessed inside or outside of the function where as local variable can be used only inside the function. We can access a global variable by declaring variable as global.

10. Output
Value of a is 45
Changed global a to 5
Value of a is 5

11. Firstly call the function
f1( )
f2( )
Then output will be look like as:
Value of n 77
Value of n 77

12. def EvenOdd():
for i in range(5):
num = int(input("Enter a number: "))
if num % 2 == 0:
print("Even")
else:
print("Odd")
EvenOdd()

13. 300#100

300@200
210#200
300@210

14. Correct code is
```
def revNumber(num):
rev=0
rem=0
while num>0:
rem-num %10
rev=rev*10+rem
num=num//10
return rev
print (revNumber(1234))
```
15.
```
def EOReplace(L):
for i in range(len(L)):
if L[i]left==0:
L[i]=L[i]+1
else:
L[i]=L[i]-1
print(L)
EOReplace(L)
```
16.
```
def INDEX_LIST(L) :
indexList-[]
for a in range(0,len(L)):
if not L[a]==0:
indexList.append(a)
return indexList
print (INDEX_LIST(L))
```
17. 22 # 40 #9 # 13 #
18.
```
def Count 3 and 7(N):
C=0
for i in range(1,N+1):
if i %3==0 or 187==0:
C=C+1
print C
```
19. Corrected code
```
def Sum(N) :
s = 0
for I in range( 1, N+1 ):
S = S + I
return S
print(Sum(10))
```
20. Output
EA3n
21.
```
def execmain( )
x = input ("Enter a number")
if (abs (x)==x);
print ("You entered a positive
else: number :")
x* = -1
print ("Number made positive: ", x)
execmain( )
```
22.
```
def findname(name):
if name in phonebook:
del phonebook[name]
else:
print("Name not found")
print("Phonebook Information")
print("Name\t Phone no")
for 1 in phonebook:
print(1, "\t", phonebook[i])
```
23.
```
def Fibonacci(max):
a = 0
b =1
while a <= max:
yield a
```

QB365 TM Study Material
QUESTION BANK 365

a, b = b, a + b
defgeneratefibo (n):
for i in Fibonacci(n):
print(i)

24. def oddtoeven(L):
for i in range (len(L)):
if (L [i]% 2! = 0):
L [i] = L [i] ~ * ~ 2
print(L)

25. (a) The new string is : gOn2Ge
Or (b) [1, 2, 3]
['ONE'. 'TWO'. 'THREE']
(1: 'ONE', 2: 'TWO', 3: 'THREE')

Case Study Questions

26. (i) (C) There are four arguments in functions as:
- Required arguments/Positional arguments.
- Keyword arguments
- Default arguments
- Variable length arguments

(ii) (A) A default arguments is an arguments that assumes a default value, if a value is not provided in the function call for that argument.
(iii) (C) Required arguments are also known positional argument. These arguments are the arguments passed to a function in correct positional order.
(iv) (B) Keyword argument
(v) (B) Argument are the values provided in function call/invoke statement. A function is a block of organised and reusable code that is used to perform a single, related action.

27. (i) def keyword used to define a function.
(ii) 0 (parentheses) Placed after the function name.
(iii) A function is a block of organised and reusable code.
(iv) Parameters are defined inside parentheses of the function.
(v) Function is used to perform single, related action.

28. (i) (B) Here total is the variable whose value is fixed during the execution of the program
(ii) (C) Python provides the def keyword to define the function the function def in line 2 is terminated with colon(:).
(iii) (C) Total is defined the sum of arg I and arg 2.
(iv) (B) This function will return the value of variable total.
(v) (D) Line 7 will print the value of total which is the sum of arg 1 and arg

QB 365 Study Material
QUESTION BANK 365