Week-5, Practice, Theory

Problem 1

Question

What is the value of `l1` at the end of execution of the code given below?

```
l1 = [1,2,3,4,5,6,7,8,9]
l1[0:2] = [10,20,30,40,50]
```

(a) `[10, 20, 30, 40, 50, 1, 2, 3, 4, 5, 6, 7, 8, 9]`

(b) `[10, 20, 30, 40, 50, 3, 4, 5, 6, 7, 8, 9]`

(c) `[[10, 20, 30, 40, 50], 3, 4, 5, 6, 7, 8, 9]`

(d) `[10, 20, 30, 40, 50, 6, 7, 8, 9]`

(e) Error

Answer

(b)

Solution

Python allows you to assign new slices to replace old slices of a list in a single operation. This is called slice assignment. So in this problem element `1,2` of l1 will be replaced by `10,20,30,40,50`. Hence, option `b` is correct.

Problem 2

Question

What will be the output of the following code-snippet?

```
1 l1=[1,2,3,4,5,6,7,8,9]
2 l1[0:1]=[10]
3
4 l2=[1,2,3,4,5,6,7,8,9]
5 l2[0]=[10]
6
7 print(l1==l2)
```

(a) `True`

(b) `False`

(c) `Error`

Answer

(b)

Solution

In line 2 element `1` of `l1` will be replaced by `10` and in line 5 element `1` of l2 will be replaced by `[10]` . After execution of code, `l1` becomes `[10, 2, 3, 4, 5, 6, 7, 8, 9]` and `l2` becomes `[[10], 2, 3, 4, 5, 6, 7, 8, 9]`, so print statement will return False. Hence, option `b` is correct.

Common data for problem 3 and 4

```
def fact(n):
    if(n==0):
        return 1
    else:
        return xxxxx
```

Problem 3

Complete the following recursive function for calculating the factorial of a positive integer `n`. It is a Multiple Select Question(MSQ).

(a) `xxxxx: n*fact(n-1)`

(b) `xxxxx: (n-1)*fact(n+1)`

(c) `xxxxx: (n-1)*fact(n-1)`

(d) `xxxxx: n*fact(n+1)`

(e) `xxxxx: fact(n-1)*n`

Answer

(a), (e)

Solution

In the above code-snippet, `xxxxx` can be replaced by `n*fact(n-1)` and `fact(n-1)*n` to compute factorial of `n`.

Problem 4

How many times will the function `fact` be called for computing the factorial of 10? It is a Numerical Answer Type (NAT) Question.

Answer

11

Solution

```
def fact(n):
    global count
    count += 1
    if (n==0):
        return 1
    else:
        return n * fact(n-1)
count = 0
print(fact(10))
print(count)
```

You can see the count value is 11 after executing the above code which define the number of execution of `fact` function. So the answer is 11.

Common data for problem 5 and 6

Observe the following definitions for the function named `display_sum`. The purpose of this function is to display the sum of the numbers passed as arguments to it.

1.

```
def display_sum(a=0,b=0,c=0):
    print(a+b+c)
```

2.

```
def display_sum(a,b=0,c=0):
    print(a+b+c)
```

3.

```
def display_sum(a=0,b,c=0):
    print(a+b+c)
```

4.

```
def display_sum(a,b=0,c):
    print(a+b+c)
```

5.

```
def display_sum(a,b,c=0):
    print(a+b+c)
```

6.

```
def display_sum(a=0,b=0,c):
    print(a+b+c)
```

Problem 5

Which of the above function definitions are invalid. It is a Multiple Select Question(MSQ).

Answer

3, 4 and 6

Solution

Non-default arguments cannot follow default arguments in function definition. So option 3, 4 and 6 not following this order because of this 3, 4 and 6 are invalid.

Problem 6

Which of the above definitions print the correct answer for `display_sum(10,10)` . It is a Multiple Select Question(MSQ).

Answer

1, 2 and 5

Solution

Option 1, 2 and 5 have valid definition of function, so for `display_sum(10,10)` we are passing `a` and `b` value in function , and `c` already defined as a default value to 0 ,so these three functions print the correct output .

Problem 7

Question

Which of the following statements are correct? It is a Multiple Select Question(MSQ).

(a) If the `return` statement is not used inside the function, the function will return 0.

(b) In order to change the value of the global variable inside the function, keyword global is used.

(c) Default argument can be defined before keyword argument in the function.

(d) Functions are an effective way to reuse program codes.

(e) Function can be passed as an argument in another function.

(f) The scope of the local variable is limited to the function where it is defined.

Answers

(b), (d), (e), (f)

Solution

Option (a) is incorrect because If the `return` statement is not used inside the function, the function will return None.

option (c) is incorrect because Non-default arguments cannot follow default arguments in function definition.

Other than a and c, all options are correct.

Problem 8

Question

What does the following code-snippet print?

```
x = 10
y = 20
z = 30
def show(x,y):
    print(x + y + z)
    x = 40
    y = 50
    print(x + y + z)

show(100,300)
print(x + y + z)
```

(a)

```
60
120
60
```

(b)

```
430
120
60
```

(c)

```
430
120
120
```

(d)

```
430
430
120
```

Answer

(b)

Solution

In first print statement inside function, x will be 100 and y will be 300 and z will be 30 the, print statement print 430. Here x and y are local variable for function.

In second print statement inside function, x will be 40 and y will be 50 and z will be 30 the, print statement print 120. Here x and y are local variable for function.

In third print statement outside the function, x will be 10 and y will be 20 and z will be 30 the, print statement print 6.Here x and y are local variable for function.

Hence, Correct option is b. Here x ,y and z are global variable for function.

Problem 9

Question

Which positive integer input from the user will make this program display `264` as the output? It is a Numerical Answer Type (NAT) Question.

```
def func(x):
    return x + 1
n = int(input())
print(int(func(n / 2) * func(n + 1)))
```

Answers

21

Solution

For input value n=21 `fun(n/2)` will return 11.5 and `func(n + 1)` will return 23 after that `print(int(11.5*23))` print 264 in output.

Problem 10

Question

Given the following function description

- Returns True if it is possible to add two of the arguments to get the remaining one and False otherwise.

Example:-

- `Check(1, 2, 3)` returns True
- `Check(3, 1, 2)` returns True
- `Check(3, 2, 2)` returns False

Which of the following function definitions produces the desired effect?

(a)

```
def Check(a, b, c):
    l = [a,b,c]
    l.sort()
    return min(l) + min(l[1:]) == max(a,b,c)
```

(b)

```
def Check(a, b, c):
    if ((a + b == c) and (a + c == b) and (b + c == a)):
        return True
    return False
```

(c)

```
def Check(a, b, c):
    if (a + b == c):
        return True
    elif ((a + c == b) or (b + c == a)):
        return True
    return False
```

(d)

```
def Check(a, b, c):
    if (a + b != c):
        return False
    elif (a + c != b):
        return False
    elif (b + c != a):
        return False
    else:
        return True
```

(e)

```
def Check(a, b, c):
    l = [a,b,c]
    l.sort()
    return l[0] + l[1] == l[2]
```

Answers

(a), (c) and (e)

Solution

Options a, c and e have the correct logic to give the correct output for the above problem..

Problem 11

Question

Code

```
1  x = 10
2  y = 20
3  z = 30
4  def show(z):
5      global x , y
6      x = 40
7      y = z
8      z = 60
9  n=int(input())
10 show(n)
11 print(x + y + z)
```

Which positive integer input `n` from the user will make this program display the following output? It is a Numerical Answer Type (NAT) Question.

```
1  120
```

Answers

50

Solution

In the `show` function `x` and `y` are defined as a global variables, so the outer variable's value can be changed inside the function and reflected outside. So after the call function in line 11, `x` becomes 40 and `y` becomes `z`(passing value `n`) and `z` remains 30. So for n=50 this prints statement will print 120 in output.

Problem 12

Question

Code

```
l1 = [1,2,3,4,5]
l2 = [6,7,8,9]
### --Fill Code--
print(newlist)
```

Select the correct code to fill in the above code-snippet to print the following output. It is a Multiple Select Question.

Output

```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

(a)

```
newlist = l1 + l2
```

(b)

```
newlist = extend(l1,l2)
```

(c)

```
newlist = l1.extend(l2)
```

(d)

```
newlist.extend(l1,l2)
```

(e)

```
l1.extend(l2)
newlist = l1
```

Answer

(a) , (e)

Solution

Here option a and e code statement is correct to print the above output.

Problem 13

Question

Code

```
def fun(l):
    l.append(4)
    return l
l=[1,2,3]
print(type(fun))
```

What does the following code-snippet print?

(a)

```
<class 'list'>
```

(b)

```
<class 'int'>
```

(c)

```
<class 'function'>
```

(d)

```
Error
```

Answer

(c)

Solution

`type(function_name)` is returns `<class 'function'>`

Problem 14

Question

Code

```
def fun(l):
    l=l*5
    return l
l=[1]
print(type(fun(l)))
print(fun(l))
```

What does the following code-snippet print?

(a)

```
<class 'list'>
[5]
```

(b)

```
<class 'list'>
[1, 1, 1, 1, 1]
```

(c)

```
<class 'int'>
5
```

(d)

```
<class 'function'>
[1, 1, 1, 1, 1]
```

Answer

(b)

Solution

`type(function_name())` returns the type of return value by function. If the `return` statement is not used inside the function, the function will return None. Here function returns a list so first print statement print `<class 'list'>` and when we multiply any list by n then all elements of list repeated n times. So second print statement will print [1, 1, 1, 1, 1].

##