ASSIGNMENT PART-A

Namespaces and Scope of Variables in Python:

1. Define a namespace in Python.
2. Explain the concept of scope in Python.
3. What is a global variable in Python?
4. Provide an example of a global variable in Python.
5. How can you access a global variable within a function in Python?
6. Describe the scope of a global variable in Python.
7. What is a local variable in Python?
8. Give an example of a local variable in Python.
9. How do you define a local variable within a function in Python?
10. Explain the scope of a local variable in Python.
11. What is the LEGB rule in Python?
12. Describe the order of variable lookup in the LEGB rule.
13. How does Python determine the scope of a variable?
14. Can you modify a global variable within a function in Python?
15. Provide an example demonstrating the modification of a global variable within a function.

ASSIGNMENT PART-B

Functions, Modules, Packages, and Libraries:

1. Define a module in Python.
2. Give an example of a Python module.
3. What is the purpose of using modules in Python?
4. Explain how to import a module in Python.
5. Describe the difference between 'import module' and 'from module import *' in Python.
6. How do you create a package in Python?
7. Provide an example of a Python package.
8. What is the purpose of using packages in Python?
9. Define a library in Python.
10. Give an example of a Python library.
11. How do you install external libraries in Python?
12. Explain the concept of a built-in function in Python.
13. Provide an example of a built-in function in Python.
14. What is the purpose of using built-in functions in Python?
15. Define a user-defined function in Python.
16. Give an example of a user-defined function in Python.
17. How do you call a user-defined function in Python?

ASSIGNMENT PART-C

25 questions focused on predicting the output types from the topics of functions, arguments, and variable scope in Python:

Question 1

```
def add(a, b):
    return a + b

result = add(3, 5)
print(result)
```

Question 2

```
def multiply(x, y=2):
    return x * y

result = multiply(4)
print(result)
```

Question 3

```
def greet(name):
    message = "Hello, " + name
    return message

print(greet("Alice"))
```

Question 4

```
def calculate(a, b=3, c=5):
    return a + b * c

result = calculate(2)
print(result)
```

Question 5

```
x = 10

def func():
    x = 5
    return x

print(func())
```

Question 6

```
def outer():
    x = 10
    def inner():
        nonlocal x
        x += 5
        return x
    return inner()

print(outer())
```

Question 7

```
def outer():
    x = 10
    def inner():
        global x
        x += 5
        return x
    return inner()

print(outer())
```

Question 8

```
def func(a, b, c=5):
    return a + b * c
result = func(2, 3)
print(result)
```

Question 9

```
def power(x, n=2):
    return x ** n

print(power(3))
```

Question 10

```
def func(a=1, b=2, c=3):
    return a * b + c

result = func(b=5)
print(result)
```

Question 11

```
x = 5

def func(x):
    x += 2
    return x

print(func(3))
print(x)
```

Question 12

```
def func(a):
    a.append(4)

lst = [1, 2, 3]
func(lst)
print(lst)
```

Question 13

```
def func(x):
    x = 10

x = 5
func(x)
print(x)
```

Question 14

```
def func():
    global x
    x = 10

func()
print(x)
```

Question 15

```
def func(a, b):
    return a + b

result = func(3, 4)
print(result)
```

Question 16

```
def func(x):
    return x * 2

result = func("Hello")
print(result)
```

Question 17

```
def func(x, y):
    return x - y
```

```
result = func(y=3, x=5)
print(result)
```

Question 18

```
def func(a, b):
    a += 2
    b += 3
    return a * b

x = 2
y = 3
result = func(x, y)
print(result)
print(x, y)
```

Question 19

```
def func(a=2, b=3):
    return a * b

result = func(b=5)
print(result)
```

Question 20

```
def func(a, b):
    return a / b

result = func(10, 3)
print(result)
```

Question 21

```
def func(x):
    x += 5
    return x

x = 2
result = func(x)
print(x)
```

Question 22

```
def func(x):
    x += "world"
    return x

result = func("Hello ")
print(result)
```

Question 23

```
def func(x, y):
    return x + y

result = func(3, "world")
print(result)
```

Question 24

```
def outer():
    x = 10
    def inner():
        nonlocal x
        x += 5
        return x
    inner()
    return x

print(outer())
```

Question 25

```
def func():
    global x
    x = 10

func()
print(x)
```

ASSIGNMENT PART-D

Here are 10 programming questions involving sequential data types like lists, tuples, dictionaries, and sets:

1. **Program Function: Merge Lists**

Write a function that takes two lists as input and returns a new list containing all the elements from both lists.

2. **Program Function: Tuple Average**

Write a function that takes a tuple of numbers as input and returns the average of those numbers.

3. **Program Function: Reverse String**

Write a function that takes a string as input and returns the reverse of that string.

4. **Program Function: Common Elements**

Write a function that takes two lists as input and returns a new list containing only the common elements between the two lists.

5. **Program Function: Dictionary Merge**

Write a function that takes two dictionaries as input and merges them into a single dictionary. If there are common keys, their values should be concatenated into a list.

6. **Program Function: Unique Elements**

Write a function that takes a list as input and returns a new list containing only the unique elements of the input list.

7. **Program Function: Set Union**

Write a function that takes two sets as input and returns a new set containing all the elements present in either set.

8. **Program Function: Sort Dictionary by Value**

Write a function that takes a dictionary as input and returns a new dictionary with the same keys but sorted by their corresponding values.

9. **Program Function: Remove Duplicates**

Write a function that takes a list as input and returns a new list with duplicates removed, preserving the original order of elements.

10. **Program Function: Count Occurrences**

Write a function that takes a list as input and returns a dictionary containing the count of occurrences of each element in the list.

ASSIGNMENT SUBMISSION DATE: 4TH APRIL 2024 THURSDAY 1:30 PM