GLS UNIVERSITY
Faculty of Computer Applications & IT
Integrated MCA

221601501 Introduction to Python

Theory Assignment Unit – 3

Q – 1 Fill in the blanks:

1. In Python, arguments to functions are passed by _____, meaning that the function receives a reference to the original object, not a copy of it.
2. When a function modifies a mutable object passed as an argument, the changes are reflected _____ the function call.
3. For immutable data types like strings and tuples, any modification within a function creates a _____ object.
4. Default argument values are specified in the function _____ and provide default values if no arguments are passed.
5. If a default argument is defined for a function, it is _____ to provide that argument when calling the function.
6. Keyword arguments are passed to a function using the _____ syntax in which the name of the parameter is followed by the value.
7. Unlike positional arguments, the order of keyword arguments in the function call _____ (does/does not) matter.
8. When using both positional and keyword arguments in a function call, positional arguments must _____ (come before/come after) keyword arguments.
9. In Python, *args allows a function to accept a variable number of _____ arguments.
10. The **kwargs syntax in a function definition allows the function to accept a variable number of _____ arguments.
11. The *args in a function call is treated as a _____, while **kwargs is treated as a _____.
12. Lambda functions in Python are also known as _____ functions because they are not bound to a name.
13. A lambda function is defined using the keyword _____, followed by the arguments, a colon, and the expression.
14. Lambda functions are typically used for _____ operations where a small, one-time use function is needed.
15. Lambda functions in Python are also known as _____ functions because they are not bound to a name.
16. A lambda function is defined using the keyword _____, followed by the arguments, a colon, and the expression.
17. Lambda functions are typically used for _____ operations where a small, one-time use function is needed.
18. The scope of a variable defines the region of the program where the variable can be _____.
19. A variable defined inside a function is called a _____ variable, while a variable defined outside any function is called a _____ variable.
20. The _____ keyword is used to declare a variable inside a function as global, allowing it to modify the variable defined in the global scope.
21. The return statement in Python functions is used to _____ a value from the function to the caller.

22. If no value is specified in a return statement, the function returns _____ by default.
23. A function in Python can return _____ (number) of values, which can be captured in multiple variables.
24. Python's built-in module for working with mathematical functions is called _____.
25. The _____ module provides functions to interact with the operating system, like creating or removing a directory.
26. To generate random numbers, the built-in module _____ is commonly used in Python.
27. To use a module in Python, you can bring it into your script using the _____ statement.
28. The import statement _____ is used to import all the names from a module directly into the current namespace.
29. If you want to import a module under a different name for convenience, you can use the syntax import module_name as _____.
30. A variable defined inside a function has a _____ scope.
31. The _____ namespace contains all the built-in names in Python.
32. If a variable is defined inside a class but outside any methods, it is considered to have _____ scope.
33. A package in Python is typically a directory that contains a special file named _____.
34. You can organize multiple related modules into a _____ to avoid naming conflicts.
35. To import a module from a package, you can use the syntax import package_name._____.

Q – 2 Answer the following:

1. Explain the difference between "Pass by Reference" and "Pass by Value" in programming languages. How does Python handle argument passing?
2. Provide examples in Python to demonstrate how mutable and immutable data types behave differently when passed to a function.
3. What are default argument values in Python? How are they used in function definitions?
4. How can default argument values improve the flexibility of function calls in Python?
5. What are keyword arguments in Python, and how do they differ from positional arguments?
6. Explain the advantages of using keyword arguments in Python functions with an example.
7. Can you mix positional and keyword arguments in a Python function call? Justify with example.
8. What are variable-length arguments in Python, and how are they implemented using *args and **kwargs?
9. What is an anonymous function (lambda function) in Python, and how does it differ from a regular function?
10. Discuss the limitations of lambda functions in Python compared to regular functions.
11. Explain the concept of scope in Python. How does Python determine the scope of a variable?
12. What are the differences between local, global, and nonlocal variables in Python? Provide examples to illustrate each type.
13. How does the global keyword affect the scope of a variable in Python?
14. What is the purpose of the return statement in Python functions? Can a function have multiple return statements?
15. What are built-in modules in Python, and why are they important?
16. Explain how the sys module can be used to interact with the Python runtime environment.
17. What is the purpose of the import statement in Python?

18. Describe the difference between importing a module and importing specific functions or classes from a module.
19. Explain the difference between import module and from module import function.
20. What are namespaces in Python, and how do they work?
21. What is a package in Python, and how does it differ from a module?
22. What is the role of the __init__.py file in a Python package?

NOTE: Students must attempt questions as below:

Enrolement Nos	Questions
202202519010001 to 202202519010015	1, 8, 11, 21
202202519010016 to 202202519010031	2, 9, 12, 22
202202519010032 to 202202519010062	3, 10, 13, 21
202202519010063 to 202202519010094	4, 14, 18, 22
202202519010095 to 202202519010125	5, 15, 19, 21
202202519010126 to 202202519010155	6, 16, 20, 22
202202519010156 to 202202519010165	7, 17, 20, 21