# 1BPLC105B/205B PYTHON PROGRAMMING

## Module-4: *Modules, Mutable vs Immutable, Object Oriented Programming*

***Part A – 5 Marks Questions***

***Modules***

1. Explain the purpose of modules in Python. How do modules improve code reusability?
2. Write a Python program to generate random integers and floating-point numbers using the random module.
3. Explain any five functions available in the Python math module with examples.
4. Describe the use of the time module in Python. Write a program to display current system time.
5. Differentiate between built-in modules and user-defined modules in Python.
6. Explain the steps involved in creating and importing a user-defined module.
7. What are namespaces in Python? Explain with a suitable example.
8. Explain scope and lookup rules in Python using LEGB rule.
9. Write short notes on: **Global scope, Local scope**
10. Explain the role of the dot (.) operator in Python modules and objects.
11. Discuss the three variants of import statements in Python with examples.
12. Differentiate between:
    - import module
    - from module import *
    - from module import function
13. Write a Python program using the math module to calculate square root, factorial, and power of a number.
14. Explain aliasing in import statements with suitable examples.
15. Write a program to generate a random password using Python modules.

---

***Mutable versus Immutable and Aliasing***

16. Define mutable and immutable objects in Python with examples.
17. Differentiate between mutable and immutable data types.
18. Explain aliasing in Python with an example.
19. What happens when mutable objects are passed as arguments to functions?
20. Write a short note on object identity and reference in Python.
21. Explain shallow copying and aliasing with examples.
22. Write a Python program demonstrating mutable list behavior.
23. Explain why strings are immutable in Python.
24. Discuss the effect of modifying aliased lists in Python.
25. Compare tuples and lists with respect to mutability.

---

***Object Oriented Programming***

26. Define a class and object in Python with syntax.
27. Explain the process of creating objects in Python classes.
28. What are attributes in a Python class? Explain instance attributes with examples.
29. Explain methods in Python classes with suitable examples.
30. Write a program to create a Student class with attributes and methods.
31. Explain the purpose of the self parameter in Python classes.
32. How are instances used as arguments and parameters in Python?
33. Explain how objects can be converted into strings using __str__() method.
34. Write a Python program demonstrating constructor and instance variables.
35. Explain how instances can be returned from methods with examples.
36. Differentiate between class variables and instance variables.
37. Write a short note on:
    - Constructor
    - Destructor

38. Explain object-oriented programming concepts supported by Python.
39. What is encapsulation? Explain with an example.
***40.*** Write a Python program to demonstrate object creation and method calling.

---

***Part B – 10 Marks Questions***

***Modules***

1. Explain the concept of modules in Python. Discuss different types of modules with suitable examples.
2. Describe the working of the random, math, and time modules with example programs.
3. Explain the procedure for creating user-defined modules and importing them into another program.
4. Discuss namespaces and scope resolution in Python using the LEGB rule with examples.
5. Explain the three import statement variants in Python. Compare their advantages and disadvantages.
6. Write a Python program using:
    - random module for OTP generation
    - time module for displaying execution time
    - math module for mathematical calculations
7. Explain attributes and the dot operator in Python modules and classes with suitable examples.
8. Write a detailed note on scope and lookup rules in Python functions and modules.

---

***Mutable versus Immutable and Aliasing***

9. Differentiate mutable and immutable objects in Python with suitable examples and diagrams.
10. Explain aliasing in Python. Discuss its advantages and disadvantages with examples.
11. Write a Python program demonstrating the behavior of mutable and immutable objects during function calls.
12. Explain the concept of object references, aliasing, and copying in Python with examples.
13. Compare lists, tuples, strings, and dictionaries based on mutability and memory behavior.
14. Explain how Python handles mutable objects internally with suitable illustrations.

---

***Object Oriented Programming***

15. Explain classes and objects in Python with syntax and examples.
16. Write a Python program to create a BankAccount class with deposit, withdraw, and balance display methods.
17. Discuss attributes and methods in Python classes with suitable examples.
18. Explain instance variables, class variables, and methods with an example program.
19. Write a Python program to demonstrate:
    - Creating a class
    - Adding methods
    - Creating objects
    - Accessing attributes
20. Explain how instances are used as arguments and return values in Python methods with examples.
21. Describe the role of constructors and the self parameter in Python object-oriented programming.
22. Explain the use of **__str__()** method in converting objects into strings with example programs.
23. Write a Python program for a Rectangle class to calculate area and perimeter using methods.
24. Develop a Python program using classes and objects to maintain student details and display results.
25. Explain the advantages of object-oriented programming in Python with suitable examples.
26. Compare procedural programming and object-oriented programming in Python.
27. Write detailed notes on:
    - Encapsulation
    - Reusability
    - Modularity in OOP
***28.*** Design and implement a Python class for employee payroll management with appropriate methods and attributes.