

USN

Course Code **1BPLC105/205B**

First / Second Semester B.E. Degree Examinations, July 2026

PYTHON PROGRAMMING

Duration: 3 hrs

Max. Marks: 100

Note: 1. Answer any FIVE full questions choosing ONE Full Question from each Module
 2. Missing data, if any, may be suitably assumed

<u>Q. No</u>	<u>Question</u>	<u>Marks</u>	<u>(RBTL:CO: PI)</u>
<u>Module-1</u>			
1.	a. State the role of type conversion in Python and list the commonly used type conversion functions.	08	(2 :1: 1.3.1)
	b. Develop a program to read the name and year of birth of a person. Display whether the person is a senior citizen or not.	06	(2 :1: 2.1.2)
	c. List the different Python data types and indicate their typical use with suitable examples.	06	(2 :1: 1.3.1)
(OR)			
2.	a. Outline how a Python program is executed and how errors are detected during execution with suitable examples.	08	(2 :1: 1.3.1)
	b. Explain the use of break and continue statements.	06	(2 :1: 1.3.1)
	c. Develop a Python program to generate the Fibonacci sequence of length N, where the value of N is read from the console.	06	(2 :1: 2.1.2)
<u>Module-2</u>			
3.	a. Illustrate the role of list operations such as append(), pop(), and slicing in data manipulation.	08	(2 :2: 1.3.1)
	b. Differentiate between aliasing and cloning of lists.	06	(2 :2: 1.3.1)
	c. Create a Python program to read N numbers from the console, form a list, and compute the mean, variance, and standard deviation with suitable messages.	06	(2 :2: 2.1.2)
(OR)			
4.	a. Differentiate between tuples, lists, and strings in Python using suitable examples.	08	(2 :2: 1.3.1)
	b. Write a Python program to read marks of six subjects from the keyboard and sort them in descending order using the Bubble Sort technique.	06	(2 :2: 2.1.2)
	c. Differentiate between indexing and slicing operations on strings in Python with suitable examples.	06	(2 :2: 1.3.1)
<u>Module-3</u>			
5.	a. Demonstrate aliasing and copying in Python dictionaries using suitable programs.	10	(2 :3: 2.1.2)

- b. Develop a Python program to read the contents of a text file, sort the lines using appropriate string and list methods, and write the sorted contents into a separate text file. **10** (2 :3: 2.1.2)

(OR)

6. a. Illustrate NumPy arrays and demonstrate array shape, slicing, and broadcasting with examples. **10** (2 :3: 1.2.1)
 b. Outline file handling in Python and demonstrate the steps involved in reading from and writing to files with suitable examples. **10** (2 :3: 1.2.1)

Module-4

7. a. Outline namespaces and variable scope in Python and demonstrate the LEGB rule with appropriate examples. **10** (2 :4: 2.1.2)
 b. Define class and object, and demonstrate the use of attributes and methods in Python using a suitable program. **10** (2 :4: 2.1.2)

(OR)

8. a. Explain the purpose of Python libraries and discuss the usage of math and random modules. **08** (2 :4: 1.2.1)
 b. Write a Python program that uses the time module to measure execution time of a loop. **06** (2 :4: 2.1.2)
 c. Develop a Python program to accept multiple students' names and marks, store them using suitable data structures (list of tuples or dictionaries), and display summary reports such as class average and topper details. **06** (2 :4: 2.1.2)

Module-5

9. a. Develop a function named DivExp which takes TWO parameters a, b, and returns a value c ($c=a/b$). Write a suitable assertion for $a>0$ in the function DivExp and raise an exception for when $b=0$. Develop a suitable program that reads two console values and calls the function DivExp. **08** (2 :5: 2.1.2)
 b. Demonstrate polymorphism in Python using a suitable example. **06** (2 :5: 2.1.2)
 c. Illustrate how exceptions are handled in Python. **06** (2 :5: 2.1.2)

(OR)

10. a. Differentiate between object mutability and copying in Python using examples. **08** (2 :5: 2.1.2)
 b. Explain the concept of object identity (sameness) in Python with examples. **06** (2 :5: 2.1.2)
 c. Compare pure functions and modifiers in Python using appropriate examples. **06** (2 :5: 2.1.2)

*** ** *