Roll No. | | | | | | | | | | | | **Total No. of Pages : 02**

Total No. of Questions : 07

BCA/B.Sc.(IT) (Sem.–3)

PROGRAMMING IN PYTHON

Subject Code : UGCA1914

M.Code : 78180

Date of Examination : 13-06-2024

Time : 3 Hrs. **Max. Marks : 60**

INSTRUCTIONS TO CANDIDATES :

1. **SECTION-A is COMPULSORY consisting of TEN questions carrying TWO marks each.**
2. **SECTION-B contains SIX questions carrying TEN marks each and students have to attempt any FOUR questions.**

SECTION-A

1. Write briefly :

a. Define the scope and lifetime of variables in Python.

b. What are three key features of Python?

c. How do you set up environment variables for Python?

d. Can you name three Python keywords?

e. Explain the purpose of the write(), tell(), and seek() methods in Python file operations.

f. What are the advantages of using functions in Python?

g. How do you define a dictionary in Python?

h. How do you iterate over a sequence in Python?

i. Explain pass by value and pass by reference in Python.

j. What is file encoding, and why is it important?

SECTION-B

2. Explore the versatility of lists in Python, explaining how they can store heterogeneous data and be manipulated using various built-in functions such as append(), extend(), and remove(). Provide examples to demonstrate list operations and list comprehensions.

3. Describe the features of Python, explaining how they contribute to its popularity and versatility as a programming language?

4. Compare and contrast tuples and lists in Python, highlighting their similarities and differences in terms of mutability, syntax, and use cases. Discuss scenarios where tuples are preferred over lists and vice versa.

5. Discuss the concept of exception handling in Python, explaining how it allows programmers to gracefully handle errors and exceptions that occur during program execution? Describe the try-except block structure and demonstrate its usage with examples of handling different types of exceptions, including built-in and custom exceptions.

6. Illustrate the concept of functions in Python, highlighting their role in code organization, reusability, and abstraction. Explain how functions can improve code readability and maintainability, and provide examples to illustrate their advantages in different scenarios?

7. Explain Object-Oriented Programming (OOP) in Python, including its key concepts such as classes, objects, inheritance, encapsulation, and polymorphism. Discuss the process of designing classes and creating objects in Python, including defining attributes and methods within classes and instantiating objects from class blueprints. Provide examples illustrating the usage of classes and objects in Python programming.