Python Programming
Module 2 : Strings, Tuples, Lists, Dictionaries

Question Bank

Part A – 5 Marks Questions

Strings

1. What is compound data type? Explain how to work with parts of a string. Explain len() function with example.
2. Explain string traversal using the for loop with an example.
3. What are string slices? Explain with suitable examples.
4. Differentiate between mutable and immutable objects in Python.
5. Explain the use of “in” and “not in” operators in strings.
6. Write a short note on the find() method in strings.
7. Explain the split() method with an example.
8. What is string comparison? Explain with examples.
9. Explain the purpose of the format() method in strings.
10. Write a Python program to count the number of vowels in a string.
11. Explain optional parameters in string functions.
12. Write a short note on cleaning up strings using strip().
13. Explain how looping and counting are performed in strings.

Tuples

1. Define tuples. How are tuples different from lists?
2. Explain tuple assignment with an example.
3. Describe how tuples are used for grouping data.
4. Explain tuples as return values with a suitable example.
5. Write a Python program to swap two variables using tuple assignment.
6. Explain tuple indexing and slicing.

Lists

1. Explain list values and accessing list elements.
2. Describe list membership operators with examples.
3. Explain list operations supported in Python.
4. What are list slices? Explain with examples.
5. Explain why lists are called mutable objects.
6. Write a note on list deletion methods.
7. Explain aliasing in lists with an example.
8. Differentiate between aliasing and cloning of lists.
9. Explain nested lists with an example.
10. Write a Python program to find the largest element in a list.
11. Explain list methods such as append(), insert(), and remove().
12. Describe the use of range() with lists.
13. Explain matrices using nested lists.
14. Differentiate between pure functions and modifiers in lists.
15. Explain objects and references in Python lists.
16. Write a Python program to reverse a list.
17. Explain how strings and lists are related in Python.

Dictionaries

1. Define dictionaries and explain dictionary operations.
2. Explain dictionary methods with suitable examples.
3. What is aliasing and copying in dictionaries?
4. Write a Python program to count word frequency using a dictionary.
5. Differentiate between lists and dictionaries.
6. Explain dictionary traversal using loops.
7. Write a short note on dictionary keys and values.
8. Explain the get() and items() methods in dictionaries.

Part B – 10 Marks Questions

Strings

1. Explain the following string operations with suitable examples:
    - Traversal
    - Slicing
    - String comparison
    - Membership operators
2. Discuss string immutability in Python. Explain various string manipulation methods with examples.
3. Explain the working of the following string methods with programs:
    - find()
    - split()
    - strip()
    - format()
4. Write a Python program to:
    - Count vowels, consonants, digits, and spaces in a string.
    - Display the frequency of each character.
5. Explain looping and counting in strings with suitable Python examples.
6. Discuss string formatting techniques in Python with examples.
7. Explain optional parameters in string functions and methods with suitable examples.
8. Write a Python program to check whether a given string is a palindrome and explain the logic.

Tuples

1. Explain tuples in Python with examples. Discuss:
    - Tuple assignment
    - Tuple indexing and slicing
    - Tuples as return values
2. Compare lists and tuples in Python with suitable examples.
3. Write Python programs using tuples for:
    - Swapping values
    - Returning multiple values from functions
    - Grouping student records
4. Explain tuple operations and tuple packing/unpacking with examples.

Lists

1. Explain list operations in Python with suitable examples.
2. Discuss list mutability and explain the concepts of:
    - Objects and references
    - Aliasing
    - Cloning
3. Explain list methods in Python with examples.
4. Write a Python program to perform the following list operations:
    - Insertion
    - Deletion
    - Searching
    - Sorting
    - Reversing
5. Explain nested lists and matrices in Python with suitable examples.
6. Discuss pure functions and modifiers in lists with examples.
7. Explain list traversal using loops and list comprehensions.
8. Write Python programs for:
    - Matrix addition
    - Matrix multiplication using nested lists.
9. Explain the relationship between strings and lists in Python with examples.
10. Discuss list slicing and membership operations with examples.
11. Write a Python program to:
    - Find the largest and smallest elements in a list
    - Remove duplicate elements
    - Sort the list in ascending order

Dictionaries

1. Explain dictionary operations and methods with suitable examples.
2. Discuss aliasing and copying in dictionaries with examples.
3. Write a Python program to create a student database using dictionaries and perform:
    - Insert
    - Update
    - Delete
    - Search operations
4. Explain dictionary traversal techniques with examples.
5. Compare lists, tuples, and dictionaries in Python.
6. Explain dictionary methods such as:
    - keys()
    - values()
    - items()
    - get()
    - update()
7. Write a Python program to count the occurrence of words in a paragraph using dictionaries.
8. Explain the advantages and limitations of dictionaries in Python.
9. Discuss the concepts of keys, values, and key-value pairs in dictionaries with examples.

****