QUESTION BANK

Unit 1:

- Explain the steps involved in the Python programming cycle.
- How does the Python interpreter execute a Python program?
- Discuss the importance of planning, coding, testing, debugging, and maintenance in the Python programming cycle.
- What is an Integrated Development Environment (IDE)? Name some popular Python IDEs.
- Discuss the features of a Python IDE that facilitate coding, debugging, and testing.
- Explain how to set up a Python project in an IDE and execute Python programs.
- Describe different ways to interact with Python programs (e.g., command-line, file input/output, user input).
- How can you pass command-line arguments to a Python program?
- Explain how to read input from the user and display output in Python programs.
- Discuss the main elements of Python programming language, including variables, data types, control structures, functions, and modules.
- Provide examples illustrating the use of variables, data types, loops, conditional statements, functions, and modules in Python.
- What is type conversion in Python? Why is it necessary?
- Explain the functions int(), float(), str(), and bool() in Python and how they are used for type conversion.
- Provide examples demonstrating type conversion between different data types in Python.
- Define expressions and assignment statements in Python.
- Discuss the arithmetic operators (+, -, *, /, //, %, **) and their precedence in Python.
- Provide examples illustrating the use of expressions, assignment statements, and arithmetic operators in Python programs.
- Explain the concept of operator precedence in Python.
- Provide a table showing the precedence of different operators in Python.
- How can parentheses be used to override operator precedence in expressions?
- Define boolean expressions and boolean data type in Python.
- Discuss the boolean operators (and, or, not) and their usage in boolean expressions.
- Provide examples demonstrating the evaluation of boolean expressions and their importance in control structures like if statements and loops.
- Explain the packing and unpacking of tuples in Python.
- Write a program to find the number of occurrences of each letter present in the string entered
by a user and store it in the dictionary.
- Explain 5 inbuilt functions of List, Tuple, Set, and Dictionary.
- Differentiate between list and tuple.
- Explain the logical and relational operators in Python.
- Write the difference between mutable and immutable data types in python.
-

Unit 2:

- Write a python program to find 2 largest elements from a list having 10 integer elements.
- Differentiate between for and while loop.
- Explain a nested loop.

Unit 3:

- Write a function called sum_digits that is given an integer num and returns the sum of the digits of num.
- Write a function called primes that is given a number n and returns a list of the first n primes. Let the default value of n be 100.
- Write a Python function that accepts a string and counts the number of upper- and lower-case letters.
- Explain the difference between local variables and global variables in functions using examples.
- A function in python can return multiple values. Give an example for it.
- Explain the difference between user defined functions and predefined functions using examples.
- What is the difference between function definition and function calling?
- Write a function that checks whether a passed string is a palindrome or not. Note: A palindrome is a word, phrase, or sequence that reads the same backward as forward, e.g., madam or nurses run.
- Create a function that takes a string as an argument and returns the most common character in that string.
- Explain Lambda function using an example.
- Elaborate the uses of lambda functions using suitable examples.
- Differentiate between lambda function and a regular function.
- Given a list of numbers [34,2,5,6,8,9,33,1,7,44]. Filter the even numbers from the list using lambda function.
- Double each and every element of the above list using lambda function.
- Write a Python program to capitalize the first and last character of each word in a string.
- Write a Python program to check whether the string is Symmetrical or Palindrome.
- What is the difference between the find() method and the index() method in Python string operations?
- Explain the concept of string slicing in Python with an example.
- How do you convert a string to uppercase and lowercase in Python?
- Explain the difference between single (''), double ("") and triple quotes (''' or """) in Python string literals.
- Explain how the strip() function works in Python string manipulation.
- What function can you use to check if a string contains only alphabetic characters in Python?
- Describe the purpose of the split() function in Python and provide an example
- How does the join() function work in Python string manipulation, and when would you use it?
- What does the find() function return when the substring is not found in the string?

- Discuss the concept of lambda functions in Python with examples using map function
- What are the types of arguments in function
- Explain Scope Rules.
- Calculate the length of a string without using a predefined function.
- Explain *args and *kwargs arguments in function.
- Write the difference between mutable and immutable data types in python.

Unit4:

- Write down the different operations performed in file handling.
- Describe Inheritance with its advantages and disadvantages
- Explain different types of constructors in Python
- Discuss the concept of class and Object with its syntax.
- Write a code to create a text file in Python. Describe the Python program to write the number
of digits and alphabets in a given input string into a File object.

Unit 5:

- Compare and contrast at least three different types of plots supported by Matplotlib (e.g., histogram, scatter plot, bar graph) and discuss scenarios where each type of plot is most suitable for data representation.
- Explain the concept of array creation in NumPy. Discuss at least three different methods for creating arrays and provide examples for each method.
- Explore the role of Pandas in data analysis. Explain the concept of a Pandas Series and discuss its advantages over traditional Python lists. Provide examples to demonstrate the creation and manipulation of Pandas Series.
- What are the advantages of using Pandas over traditional data structures like lists or dictionaries in Python?
- Write a Python function that takes two NumPy arrays as input and returns their element-wise sum. Test your function with arrays [1, 2, 3] and [4, 5, 6].
- Explain how you can transpose a NumPy array using both the .T attribute and the np.transpose() function.
- What is Matplotlib in Python? How can you create a histogram in Matplotlib?
- What types of plots can you create with Matplotlib? Provide examples of at least three different plot types.