College of Computer Application (04)
Silver Oak College of Computer Application
Bachelor Of Computer Application
Department of Computer Application (023)

Semester :	4	**Academic Year :**	2023-24
Subject Name :	Programming in Python	**Subject Code :**	2040233211/2040233205

QUESTION BANK

1 Describe the history and evolution of Python.

2 List 5 major features of Python and their benefits.

3 Mention 3 areas where Python is widely used.

4 Explain the steps to install Python on Windows.

5 What is the role of indentation in Python and how does it differ from other languages?

6 Write a program to print the sum of 2 user-input numbers.

7 Differentiate between variables and constants in Python.

8 Provide examples of valid and invalid identifiers in Python.

9 Explain the different types of literals available in Python with examples.

10 What is the purpose of comments in Python code?

11 Compare and contrast arithmetic, comparison, and logical operators in Python.

12 Demonstrate user input and output in Python.

13 Explain implicit and explicit type conversion in Python.

14 Write a program to calculate the area of a triangle using user inputs.

15 What is the use of the break statement and when should it be used?

16 Differentiate between conditional and looping statements.

17 Explain the syntax and functionality of the if statement with examples.

18 How is the else statement used with the if statement?

19 Write a program to check if a given number is even or odd.

20 What is the purpose of the while loop in Python? Give an example.

21 Explain the difference between a pre-test and a post-test loop.

22 Write a program to print the first 10 natural numbers using a for loop.

23 What is the use of the else and continue statements within loops?

24 Explain string indexing and slicing in Python.

25 Demonstrate common string operations like concatenation, searching, and replacement.

26 Explain built-in methods for modifying and manipulating strings in Python.

27 Explain string formatting and format specifiers in Python.

28 Write a program to display a formatted message with user-provided name and age.

29 Compare and contrast mutable and immutable objects in Python.

30 What are docstrings and how are they used in Python code?

31 Explain different data structures: lists, dictionaries, sets, and tuples.

32 Demonstrate creating, accessing, and modifying elements in a list.

33 Explain list comprehensions and their simplification of list creation.

34 Define key-value pairs in a dictionary and demonstrate access and modification.

35 Differentiate between lists and dictionaries.

36 Explain sets in Python and their unique properties.

37 Differentiate between sets and dictionaries.

38 Compare and contrast mutable and immutable data structures in Python.

39 Define a function in Python and explain its scope and argument types.

40 Explain positional and keyword arguments in function calls.

41 What are *args and **kwargs in Python functions and how are they used?

42 Explain packing and unpacking arguments and values in Python.

43 Demonstrate writing and utilizing Lambda functions in Python.

44 What is recursion and how can it be implemented in Python?

45 Explain the importance of exception handling in Python and common exception types.

46 Demonstrate using try-except blocks to handle different exceptions.

47 What is the purpose of the finally block in exception handling?

48 Write a program to calculate the factorial of a number with exception handling for invalid input.

49 Explain the concept of file handling in Python.

50 Demonstrate opening, reading, writing, and closing files in Python.

51 What is the purpose of the with statement in file handling?

52 Explain Python modules and how to import them into your code.

53 Name some commonly used Python modules and their functionalities.

54 Explain the use of regular expressions in Python and give examples.

55 Demonstrate reading and writing dates and times using the datetime module.

56 Define the basic building blocks of OOP in Python.

57 Define a class and object in Python with an example.

58 Explain inheritance in Python and its role in code reuse.

59 What is encapsulation and how is it achieved in Python classes?

60 Explain polymorphism and its implementation in Python.

61 What is data abstraction and its relation to class design in Python?

62 Explain the use of iterators and generators in Python with examples.

63 What are decorators in Python and how are they used?

64 Write a simple program demonstrating a class with attributes, methods, and inheritance.

65 Briefly introduce MySQL and its role as a relational database management system.

66 Explain the steps to install and configure MySQL in your environment.

67 Demonstrate establishing a database connection in Python using a library like pymysql.

68 Explain creating, selecting, and modifying tables in MySQL through Python code.

69 Implement INSERT, UPDATE, and DELETE operations on a table using Python and SQL.

70 What is the purpose of commit and rollback operations in database transactions?

71 Explain how to handle errors and exceptions while working with databases in Python.

72 Write a Python program to connect to a MySQL database, retrieve data from a table, and display it on the screen.

73 Discuss the benefits and drawbacks of using Python for web development.

74 Briefly explain the concept of data analysis and how Python can be used in this field.

75 Give an example of a machine learning application built using Python.

76 Compare and contrast Python with other popular programming languages like Java and C++.

77 Discuss the importance of unit testing and code documentation in Python projects.