Department of Information Technology

B. Tech. Mid Question Bank (R22 Regulation)

Academic Year: 2024-2025 **Semester: III**

Subject Name: Programming With Python

Faculty Name: Y. Satyam

PART-A

Q.No	Questions	Marks	BL	CO	Unit No
1	What is the purpose of the `input()` function in Python?	2	L1	CO1	I
2	Explain the difference between `=` and `==` in Python.	2	L1	CO1	I
3	What is the significance of indentation in Python programming?	2	L2	CO1	I
4	Write a simple Python program to print "Hello, World!" to the console.	2	L1	CO1	I
5	What are the two types of conditional statements available in Python?	2	L2	CO1	I
6	Describe the function of the `break` statement in a Python loop.	2	L2	CO1	I
7	What is the primary difference between lists and arrays in Python?	2	L1	CO2	II
8	How do you create a NumPy array in Python? Provide a simple example.	2	L1	CO2	II
9	What is the purpose of the reshape() method in NumPy?	2	L2	CO2	II
10	Explain how to perform element-wise addition of two NumPy arrays.	2	L1	CO2	II
11	Describe the use of the flatten() method in NumPy arrays.	2	L1	CO2	II
12	How do you convert a string to uppercase in Python? Provide an example.	2	L2	CO2	II
13	What is the purpose of the return statement in a Python function?	2	L2	CO3	III
14	How do you define a recursive function in Python? Provide a brief example.	2	L2	CO3	III
15	Explain how to create a list using the range() function in Python.	2	L2	CO3	III
16	What is the difference between a tuple and a list in Python?	2	L2	CO3	III
17	How do you sort the elements of a dictionary by its values using a lambda function in Python?	2	L2	CO3	III
18	Write a Python code snippet to convert a list of tuples into a dictionary.	2	L2	CO3	III
19	What is the purpose of the open() function in Python?	2	L1	CO4	IV
20	List two built-in methods for file objects in Python.	2	L2	CO4	IV
21	How do you raise an exception in Python?	2	L1	CO4	IV
22	What is the purpose of the assert statement in Python?	2	L1	CO4	IV
23	How do you import a module in Python?	2	L2	CO4	IV
24	What is a namespace in Python?	2	L1	CO4	IV
25	What is a class in Python?	2	L1	CO5	V
26	Define inheritance in the context of Object-Oriented Programming (OOP).	2	L1	CO5	V
27	What is the purpose of regular expressions in Python?	2	L2	CO5	V

28	Name two special characters commonly used in Python regular expressions.	2	L2	CO5	V
29	What is a thread in the context of Python programming?	2	L2	CO5	V
30	Briefly explain the Global Interpreter Lock (GIL) in Python.	2	L2	CO5	V

PART-B

Q.No	Questions	Marks	BL	CO	Unit No
1	Discuss the history and evolution of Python. How has it changed over the years and what are its major versions?	4	L1	CO1	I
2	Explain the different data types available in Python. Provide examples of how to declare variables of these data types.	4	L1	CO1	I
3	Write a Python program that prompts the user to enter their age and then prints a message based on whether the user is eligible to vote (18 years or older). Include the use of conditional statements in your code.	4	L2	CO1	I
4	Describe the purpose and use of the `continue` statement in Python. Illustrate its use with a sample Python program that skips even numbers in a loop.	4	L1	CO1	I
5	Explain the role of the `assert` statement in Python. Write a short code snippet that demonstrates its use in verifying that a condition holds true.	4	L2	CO1	I
6	Differentiate between the `while` and `for` loops in Python. Provide examples of each loop and explain when you might choose one over the other.	4	L2	CO1	I
7	Compare and contrast the different types of operators available in Python (arithmetic, comparison, logical, and assignment operators). Provide examples of each type and explain their use in Python programming.	8	L3	CO1	I
8	Write a Python program to demonstrate the use of `if`, `elif`, and `else` statements. The program should categorize a given integer as "Negative", "Zero", or "Positive" and also check if the number is even or odd.	8	L3	CO1	I
9	Explain the concept of control flow in Python, focusing on the `break`, `continue`, and `pass` statements. Provide a Python program that uses each of these statements within loops to demonstrate their effects.	8	L3	CO1	I
10	Explain how to create and initialize a NumPy array. Provide an example of creating a 2D array and demonstrate how to access its elements.	4	L1	CO2	II
11	Describe the process of matrix multiplication in NumPy. Write a Python code snippet to perform matrix multiplication on two 2x2 matrices.	4	L1	CO2	II
12	What are the key attributes of a NumPy array? Explain with examples how to use shape, size, and dtype attributes.	4	L2	CO2	II
13	Illustrate the difference between string slicing and string searching in Python with examples.	4	L2	CO2	II
14	How can you sort a list of strings in Python? Write a Python code snippet to sort a list of strings alphabetically.	4	L2	CO2	II
15	Demonstrate how to use the numpy.reshape() method to change the shape of an array. Provide an example with a 1D array reshaped into a	4	L2	CO2	II

	2x3 2D array.				
16	Discuss the different types of arrays available in NumPy and their applications. Compare the numpy.array() function with numpy.zeros(), numpy.ones(), and numpy.arange() in terms of their creation and initialization of arrays. Provide examples of each.	8	L3	CO2	II
17	Explain matrix operations in NumPy, including matrix addition and multiplication. Write Python code to perform these operations on two 2x2 matrices and explain the results.	8	L3	CO2	II
18	Illustrate how to perform string manipulation in Python. Discuss various string operations such as slicing, searching, and sorting. Provide Python code examples to demonstrate each operation.	8	L3	CO2	II
19	Define a Python function that takes two parameters and returns their sum. Write a function call to demonstrate how it works and explain the concept of parameters in function definitions.	4	L2	CO3	III
20	Write a recursive function to calculate the factorial of a given number. Explain how recursion works with this function and demonstrate its use with an example.	4	L2	CO3	III
21	Discuss how to perform operations on lists in Python. Provide examples for common operations such as appending, extending, and removing elements.	4	L2	CO3	III
22	Explain how to create and access elements in a tuple. Illustrate with an example and discuss how tuples differ from lists in terms of mutability.	4	L2	CO3	III
23	Write a Python function that takes a dictionary as an argument and returns a sorted list of its keys based on their corresponding values. Use a lambda function to perform the sorting.	4	L2	CO3	III
24	Convert a list of strings into a dictionary where each string is a key and its length is the value. Provide a Python code example for this conversion.	4	L2	CO3	III
25	Explain the difference between read() and readlines() methods when working with file objects in Python.	4	L2	CO4	IV
26	Describe the process and importance of using context management (with statement) when handling files in Python.	4	L2	CO4	IV
27	How can you create a custom exception in Python? Provide a brief example.	4	L2	CO4	IV
28	Discuss the role of the try, except, else, and finally blocks in exception handling.	4	L2	CO4	IV
29	Explain the difference between import module and from module import attribute in Python.	4	L2	CO4	IV
30	What are Python packages, and how do they differ from modules? Include an example of how to create and import a package.	4	L2	CO4	IV
31	Discuss the file system operations in Python, including how to navigate directories, create, rename, and delete files and directories. Provide examples demonstrating these operations.	8	L3	CO4	IV
32	Provide an in-depth explanation of the exception hierarchy in Python. Discuss the various types of built-in exceptions and how they are structured. Include examples of how to handle multiple exceptions and how to use the sys module to obtain exception details.	8	L3	CO4	IV
33	Explain the concept of modules and packages in Python. Describe how to	8	L3	CO4	IV

	create a module and a package, and demonstrate how to import and use them in a Python program. Discuss the advantages of using modules and packages for code organization and reusability.				
34	Explain the concept of polymorphism in Python with an example.	4	L2	CO5	V
35	Describe the difference between abstract classes and interfaces in Python. Provide an example of an abstract class.	4	L2	CO5	V
36	How do you use the 're' module in Python to find all matches of a pattern in a string? Provide a brief example.	4	L2	CO5	V
37	Explain the use of the '^' and '$' symbols in Python regular expressions with examples.	4	L2	CO5	V
38	Compare and contrast threads and processes in Python. Provide scenarios where each would be appropriate.	4	L2	CO5	V
39	Explain how the 'Threading' module is used to create and manage threads in Python. Provide a simple example demonstrating the creation of a thread.	4	L2	CO5	V
40	Discuss the principles of inheritance and polymorphism in Python. Provide examples to illustrate how inheritance allows for code reuse and how polymorphism enables methods to be used interchangeably across different classes.	8	L3	CO5	V
41	Explain how regular expressions are utilized for pattern matching in Python. Discuss the significance of special symbols and characters in regular expressions, providing examples for each of the following: '\d', '\w', '\s', '^', and '$'. Demonstrate how to use the 're' module to compile a regular expression and perform a search operation.	8	L3	CO5	V
42	Describe the challenges and considerations of multithreading in Python, focusing on the Global Interpreter Lock (GIL). Explain how the Threading module can be used to create and manage threads, including a discussion on thread synchronization techniques such as locks, semaphores, and condition variables. Provide code examples to illustrate these concepts.	8	L3	CO5	V