**Theory Questions for Interview Preparation**

**Introduction to Python**

1. What are the key features of Python that make it popular?
2. How do you install Python and set up the environment?
3. How can you run a Python script from the command line?
4. What is the difference between Python's interactive mode and IDEs like PyCharm or VS Code?
5. How do you write a basic Python program that prints "Hello, World!"?
6. How can you add comments in Python code?

**Variables and Data Types**

1. How do you declare and initialize variables in Python?
2. What are the core data types in Python? Explain each briefly.
3. How does Python handle type conversion between data types?
4. What is dynamic typing in Python, and how does it work?
5. Can you explain the difference between mutable and immutable data types in Python?
6. What is the difference between identifiers and reserved keywords in Python?

**Control Flow**

1. What is the syntax for an if statement in Python?
2. How does the elif statement work in Python?
3. Explain the for and while loops in Python with examples.
4. What is the purpose of the break and continue statements in loops?
5. What does the pass statement do in Python?
6. How do you create list comprehensions, and what is their advantage over using loops?

**Functions**

1. How do you define a function in Python?
2. What is the difference between positional, keyword, and default arguments in Python functions?
3. What is the purpose of the return statement in a function?
4. What are lambda functions, and how are they different from normal functions?
5. What is recursion in Python? Can you provide an example?
6. What is the scope of variables in Python, and how does the global keyword work?
7. What is the significance of \*args and \*\*kwargs in function definitions?

**Modules and Packages**

1. How do you import modules in Python?
2. What is the purpose of the \_\_init\_\_.py file in Python packages?
3. What is the difference between importing a module using import and from ... import?
4. How do you create a custom module in Python?

**File Handling**

1. How do you read and write to text files in Python?
2. What are different file modes available in Python, and what do they do (e.g., r, w, a)?
3. How can you handle CSV files in Python?
4. How do you parse and manipulate JSON data in Python?
5. What is exception handling in file operations, and how can you handle file-related exceptions?

**Error and Exception Handling**

1. What are the common types of errors in Python?
2. How do you use try, except, else, and finally blocks for error handling in Python?
3. How do you raise a custom exception in Python using raise?
4. What is the purpose of custom exceptions in Python, and how do you create one?

**Object-Oriented Programming (OOP)**

1. What is the concept of classes and objects in Python?
2. What is the purpose of the \_\_init\_\_ and \_\_del\_\_ methods in Python classes?
3. What is inheritance in Python? Explain the different types of inheritance.
4. What is method overriding in Python, and how does it work?
5. How does Python handle polymorphism?
6. What is encapsulation, and how can you achieve it in Python?

**Data Structures in Python**

1. What are the differences between lists and tuples in Python?
2. How does Python's dictionary work, and how do you access its values?
3. What are sets in Python, and how are they different from lists?
4. How do you create, manipulate, and iterate through lists, sets, and dictionaries?

**Advanced Topics**

1. What are decorators in Python? How do they work, and why are they used?
2. How do generators work in Python, and what is the purpose of the yield keyword?
3. What is the difference between an iterator and an iterable in Python?
4. Explain the concept of context managers and the with statement in Python.
5. What is the difference between multithreading and multiprocessing in Python?
6. What is Asyncio in Python, and how does it work?

**Programming Questions for Interview Preparation**

**1. Introduction to Python**

* **Q1**: Write a Python program to print "Hello, World!".
* **Q2**: Write a Python program to demonstrate the use of comments. Include both single-line and multi-line comments.

**2. Variables and Data Types**

* **Q3**: Write a Python program to swap two numbers without using a temporary variable.
* **Q4**: Write a Python program that takes two inputs (a float and an integer) from the user, converts them into their respective data types, and prints the result.
* **Q5**: Write a Python program that checks the type of a given variable.
* **Q6**: Write a Python program that converts a string to an integer and a float to a string.

**3. Control Flow**

* **Q7**: Write a Python program that checks whether a number is even or odd.
* **Q8**: Write a Python program to print all the numbers from 1 to 100 that are divisible by both 3 and 5.
* **Q9**: Write a Python program that prints the Fibonacci series up to a given number n.
* **Q10**: Write a Python program to demonstrate the use of break, continue, and pass in loops.
* **Q11**: Write a Python program to implement a simple calculator using if-elif-else statements.

**4. Functions**

* **Q12**: Write a Python function that takes two numbers as arguments and returns their sum.
* **Q13**: Write a Python function to check if a given number is a prime number.
* **Q14**: Write a Python function that finds the factorial of a number using recursion.
* **Q15**: Write a Python program that demonstrates the use of \*args and \*\*kwargs in a function.
* **Q16**: Write a Python program that demonstrates the use of a lambda function to add two numbers.

**5. Modules and Packages**

* **Q17**: Write a Python program that imports the math module and calculates the square root of a number.
* **Q18**: Write a Python program to create a custom module that contains a function for calculating the area of a circle. Import this module in your main program and use it.
* **Q19**: Write a Python program to create a package shapes that contains two modules: circle.py and rectangle.py. Each module should have a function that calculates the area of the respective shape.

**6. File Handling**

* **Q20**: Write a Python program to read a file and print its contents.
* **Q21**: Write a Python program to write a list of strings into a text file, one string per line.
* **Q22**: Write a Python program to append text to an existing file.
* **Q23**: Write a Python program that reads a CSV file and prints each row.
* **Q24**: Write a Python program that converts a Python dictionary into a JSON string and saves it to a file.

**7. Error and Exception Handling**

* **Q25**: Write a Python program that handles a division by zero error using try-except block.
* **Q26**: Write a Python program that raises a custom exception if an invalid number is provided.
* **Q27**: Write a Python program that handles multiple exceptions using try-except and displays a custom message for each exception.

**8. Object-Oriented Programming (OOP)**

* **Q28**: Write a Python class Car with attributes like make, model, and year. Create an object of the class and display its attributes.
* **Q29**: Write a Python class Rectangle that has methods to calculate the area and perimeter. Create an object and display the area and perimeter.
* **Q30**: Write a Python program that demonstrates method overriding by creating a base class Animal and derived class Dog. Override the speak method in the Dog class.
* **Q31**: Write a Python program that demonstrates multiple inheritance by creating a class Bird that inherits from classes Animal and Flyable.

**9. Data Structures in Python**

* **Q32**: Write a Python program to create a list of numbers and print the sum of all numbers.
* **Q33**: Write a Python program to remove duplicate elements from a list.
* **Q34**: Write a Python program that takes a list of numbers and returns a list containing only the even numbers.
* **Q35**: Write a Python program to create a dictionary where the keys are names and values are ages, then print the name of the oldest person.
* **Q36**: Write a Python program to merge two dictionaries into one.

**10. Advanced Topics in Python**

* **Q37**: Write a Python program to create a decorator that measures the execution time of a function.
* **Q38**: Write a Python program to create a generator that generates Fibonacci numbers.
* **Q39**: Write a Python program that uses yield to return a list of prime numbers up to n.
* **Q40**: Write a Python program that uses the with statement to open a file and automatically close it after reading.