Q1. What is the purpose of the try statement?

Ans1

The try statement is often used in conjunction with the except statement, which allows the program to handle specific types of exceptions that may be raised during the execution of the code in the try block.

he try statement is often used in conjunction with the except statement, which allows the program to handle specific types of exceptions that may be raised during the execution of the code in the try block.

Q2. What are the two most popular try statement variations?

Ans2

try-except: This variation of the try statement is used to catch and handle specific exceptions that may occur during the execution of a block of code. In this variation, one or more except blocks are included after the try block, with each except block designed to catch a specific type of exception.

try-finally: This variation of the try statement is used to ensure that a block of code is executed regardless of whether an exception occurs or not. In this variation, a finally block is included after the try block, which contains code that must be executed after the try block, regardless of whether an exception was raised or not.

Q3. What is the purpose of the raise statement?

Ans3

The raise statement allows the programmer to create and raise their own custom exceptions, or to re-raise an exception that was caught in a try-except block.

The raise statement can be used in a number of different ways. For example, it can be used to:

* Raise a built-in exception: The raise statement can be used to raise one of Python's built-in exceptions, such as ValueError, TypeError, or IndexError, when an error condition is encountered during the execution of a program.
* Raise a custom exception: The raise statement can be used to create and raise a custom exception, defined by the programmer, to handle specific errors or conditions in the program.
* Re-raise an exception: The raise statement can be used in a except block to re-raise an exception that was caught, allowing it to be handled by an outer try-except block or by the Python interpreter.

Q4. What does the assert statement do, and what other statement is it like?

Ans4

The assert statement in Python is used to test whether a condition is true, and raises an AssertionError exception if the condition is false. It can be thought of as a debugging aid that allows the programmer to verify assumptions about the program's state at a particular point in time.

where condition is the expression that is being tested, and message is an optional message that is displayed if the assertion fails.

The assert statement is similar to the if statement, in that it allows the programmer to test a condition and execute different blocks of code depending on whether the condition is true or false. However, unlike the if statement, the assert statement is typically used for testing assumptions that should always be true, rather than for making decisions based on user input or other dynamic conditions.

Q5. What is the purpose of the with/as argument, and what other statement is it like?

Ans5

The with/as statement in Python is used to define a context in which a resource is used and automatically released, even if an error occurs. It can be used with objects that support the context management protocol, such as files or network connections, to ensure that they are properly closed or released when they are no longer needed.

When the with/as statement is executed, Python evaluates the expression and calls its \_\_enter\_\_() method to acquire the resource. Then, the block of code inside the with statement is executed, using the resource as needed.