Q1. What is the purpose of the try statement?

Ans: The try statement in programming languages, including in Python, is used for exception handling. It allows you to define a block of code to be tested for errors while it is being executed. The purpose of the try statement is to help you deal with situations where you anticipate that an error might occur during the execution of a block of code. By using try with except and optionally finally blocks, you can control what happens when an error occurs and handle it gracefully, preventing the entire program from crashing.

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

Ans: 1. try….except statement 2. Try…except….else….finally statement

Q3. What is the purpose of the raise statement?

Ans: In Python, the raise statement is used to explicitly raise an exception. It allows you to throw an exception when a certain condition is met or when you want to signal that an error or unexpected condition has occurred. The raise statement is typically used in combination with the try statement for custom error handling.

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

Ans: The assert statement in Python is used as a debugging aid. It takes an expression and an optional error message. If the expression evaluates to false, it raises an AssertionError with the provided error message. It is primarily used to test the internal state of a program or to ensure that certain conditions are met.

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

Ans: The with statement in Python is used to wrap the execution of a block with methods defined by a context manager. It is particularly useful when working with unmanaged resources like files or network connections, as it ensures that the resources are properly managed and released after the block of code is executed, even if an exception occurs.

The as keyword within the with statement is used to capture the result of the context manager, typically for further use within the block of code. It assigns the result of the \_\_enter\_\_ method of the context manager to a variable, which can then be used within the block of code.