Q1. What is the purpose of the try statement?

Try: This block will test the excepted error to occur

A try statement includes keyword try, followed by a colon (:) and a suite of code in which exceptions may occur. It has one or more clauses.

During the execution of the try statement, if no exceptions occurred then, the interpreter ignores the exception handlers for that specific try statement.

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

try / except

and

try / except / except

Q3. What is the purpose of the raise statement?

The raise keyword is used to raise an exception. You can define what kind of error to raise, and the text to print to the user.

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

The assert keyword is used when debugging code. The assert keyword lets you test if a condition in your code returns True, if not, the program will raise an AssertionError. You can write a message to be written if the code returns False

Type annotations (or sometimes typing. cast ) are a more modern alternative to assert isinstance

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

In Python, with statement is used in exception handling to make the code cleaner and much more readable. It simplifies the management of common resources like file streams.

It is like try/finally statement.

In Python, the with statement replaces a try-catch block with a concise shorthand. More importantly, it ensures closing resources right after processing them. A common example of using the with statement is reading or writing to a file. A function or class that supports the with statement is known as a context manager. A context manager allows you to open and close resources right when you want to. For example, the open() function is a context manager. When you call the open() function using the with statement, the file closes automatically after you’ve processed the file.

The with statement is a replacement for commonly used try/finally error-handling statements. A common example of using the with statement is opening a file. To open and write to a file in Python, you can use the with statement as follows:

with open("example.txt", "w") as file:

file.write("Hello World!")

The with statement automatically closes the file after you’ve completed writing it.

Under the hood, the with statement replaces this kind of try-catch block:

f = open("example.txt", "w")

try:

f.write("hello world")

finally:

f.close()