Q1. What is the purpose of the try statement?
Answer: The try statement includes a block of code which is tried to check for errors. If there's no error, the try block executes successfully and skips any exception handlers specified. But if the try block returns an error , the exception handler catches the error which can be displayed as required. Abrupt termination of code doesnt occur if try and except are both specified.

Q2. What are the two most popular try statement variations?
Answer: 1) try, except, else, finally statement.
2) try, except statement
3) try, except, else statement
4) try, raise, except statement

Q3. What is the purpose of the raise statement?
Answer: Python raise Keyword is used to raise exceptions or errors. The raise keyword raises an error and stops the control flow of the program. It is used to bring up the current exception in an exception handler so that it can be handled further up the call stack.

Q4. What does the assert statement do, and what other statement is it like?
Answer: The assert statement is used to continue the execute if the given condition evaluates to True. If the assert condition evaluates to False, then it raises the AssertionError exception with the specified error message.

Another statement like assert is assertEqual() which tests the equality of two values, it takes two values and returns true or false depending on the result

Q5. What is the purpose of the with/as argument, and what other statement is it like?
Answer: The with statement is used for file handling or exception handling in python, The with statement in python ensures the the resources are freed appropriately after the execution and there's no need to call file.close() upon execution,

Example:

```
with open('file-path', 'w') as file:
    file.write('Lorem ipsum')
```

Like with as, we could also use, “try finally” block such as

out = open(txtFile, 'w') try: for line in gzipHandler: out.write(line) finally: out.close() gzipHandler.close()

Example:

```
out = open(FileName, 'w')
try:
    for i in range(0,9):
        out.write(line)
finally:
    out.close()
```