Q1. What is the purpose of the try statement?
Answer : try statement is used for Exception Handling in Python. It consists of code of statements that are checked for errors during runtime. If code within try block raises exception , then it becomes easy to revert this exception in followed except block. This try and except block in union avoids crashing of program due to exception.

Q2. What are the two most popular try statement variations?
Answer: try and except block :
The try block is used to check some code for errors i.e the code inside the try block will execute when there is no error in the program. Whereas the code inside the except block will execute whenever the program encounters some error in the preceding try block.

```
try:
   # Some Code
except:
   # Executed if error in the
   # try block
```

try , except and else clause :
The code enters the else block only if the try clause does not raise an exception.

```
try:
   # Some Code
except:
   # Executed if error in the
   # try block
else:
   # execute if no exception
```

Q3. What is the purpose of the raise statement?
Answer: raise statement is used to trigger exception, if certain condition is not as per condition requirement of programmer. It helps in triggering exception as per need of programmer and its program logic.

Q4. What does the assert statement do, and what other statement is it like?
Answer: There are few assertions that programmer always want to be true to avoid code failure. This type of requirement is fulfilled by assert statement. This statement takes a Boolean condition output of which if true, further program executes. If output of assert statement is false it raises an Assertion Error.

Q5. What is the purpose of the with/as argument, and what other statement is it like?
Answer : with/as statement simplifies use of file handling in python. When we use a with statement for file reading and/or writing it is need not to be followed by fle.close().The with statement itself ensures proper acquisition and release of resources, this avoids triggering of exception if file closing is unknowingly forgotten in the code execution.
E.g:

```
# using with statement
with open('file_path', 'w') as file:
    file.write('hello world !')
```