ASSIGNMENT - 7

Q1. What is the purpose of the try statement?

Ans: The try statement is used to handle exceptions in Python. It allows you to wrap a block of code that might raise an exception and define how to handle that exception if it occurs. The try block is followed by one or more except blocks to catch and handle specific exceptions.

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

Ans: Two popular variations of the try statement:

* try-except: The basic form of the try statement includes a try block followed by one or more except blocks. It attempts to execute the code in the try block, and if an exception occurs, it searches for a matching except block to handle that specific exception.
* try-except-finally: This variation includes an optional finally block after the try and except blocks. The code in the finally block is executed regardless of whether an exception occurred or not, allowing for cleanup operations like closing files or releasing resources.

Q3. What is the purpose of the raise statement?

Ans: The raise statement is used to explicitly raise exceptions in Python. It allows you to trigger built-in exceptions or create custom exceptions by instantiating exception classes and providing optional arguments to describe the cause of the exception.

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

Ans: The assert statement is used for debugging purposes to check if a condition is true. If the condition evaluates to False, the assert statement raises an AssertionError exception.

It's similar to the if statement but specifically focuses on detecting bugs or logical errors in the code during development. It's used to ensure that certain conditions that should always be true are indeed true.

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 for resource management, specifically to ensure proper acquisition and release of resources, such as files or network connections.

The with statement works in conjunction with the as keyword to create a context manager. It's similar to a try-except-finally block but simplifies the syntax for managing resources. For example, using with open('file.txt') as file: ensures the file is properly closed after use, even if exceptions occur within the block.