

## Python Advanced Assignment-7

### **Q1. What is the purpose of the try statement?**

**Ans.**

The ``try`` statement is used to wrap code that may potentially raise an exception. It allows the program to handle errors gracefully without terminating abruptly. If an exception occurs within the ``try`` block, the control is passed to the ``except`` block where the exception can be handled. This provides an opportunity for recovery, cleanup, or alternative behavior.

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

**Ans.**

1. ``try-except``: This is the most common variation, where code is enclosed in the ``try`` block, and if an exception occurs, the ``except`` block handles it.

```
try:
```

```
    result = 10 / 0
```

```
except ZeroDivisionError:
```

```
    print("Cannot divide by zero.")
```

2. ``try-except-finally``: In this variation, the ``finally`` block is added after the ``except`` block. It runs whether an exception occurs or not, allowing for code that must always execute (like cleanup operations).

```
```python
```

```
try:
```

```
    file = open("data.txt", "r")
```

```
except FileNotFoundError:
```

```
    print("File not found.")
```

```
finally:
```

```
    print("Cleaning up.")
```

### **Q3. What is the purpose of the raise statement?**

**Ans.**

The ``raise`` statement is used to manually trigger an exception. It allows programmers to create their own error conditions when certain conditions in the code are not met, or when they want

to signal that an error state has been reached. It can also be used to re-raise an existing exception to propagate it further.

```
if age < 0:  
    raise ValueError("Age cannot be negative.")
```

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

**Ans.**

The `assert` statement tests a condition in the code. If the condition evaluates to `False`, it raises an `AssertionError` with an optional message. It's similar to the `raise` statement because it raises an exception when a condition is not met. Assertions are commonly used for debugging purposes or to validate internal logic.

```
assert x > 0, "x must be positive"
```

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

**Ans.**

The `with/as` construct is used to manage resources, such as opening and closing files or acquiring and releasing locks. It ensures that cleanup code is executed automatically when exiting the block, even if an exception occurs. It is similar to a `try-finally` statement, as both ensure that a block of code (like closing a file) runs regardless of whether an error occurs.

```
with open("data.txt", "w") as file:  
    file.write("Hello, world!")
```

In this example, the file is automatically closed when the `with` block is exited, similar to what could be achieved with `try-finally`.