Python_advance_assignment_8

Q1. What are the two latest user-defined exception constraints in Python 3.X?

In [ ]:

```
Ans: raise and assert are the two latest user-defined exception constraints in Python
```

Q2. How are class-based exceptions that have been raised matched to handlers?

In [ ]:

```
Ans: In python, Users can define custom exceptions by creating a new class.
This exception class has to be derived, either directly or indirectly from built-in
Exception class.
This new exception class like other exceptions can be raised using the raise statement
with an optional error message.
```

Q3. Describe two methods for attaching context information to exception artefacts ?

In [ ]:

```
Ans: The process() method of LoggerAdapter is where the contextual information is adde
to the logging output.
Its passes the message and keyword arguments of the logging call, and it passes back
modified versions of these to use in the call to the underlying logger.

Other method that can be used is exception(), Logs a messgae with level ERROR on this
logger.
The arguments are interpreted as for debug(). Exception info is added to the logging
message.
```

Q4. Describe two methods for specifying the text of an exception object's error message ?

In [ ]:

```
Ans: raise and assert are two methods for specifying the text of an exception object's
error message.
raise statement is used to trigger explicit exception, if certain
condition is not as per requirement of programmer.

It helps in triggering exception as per need of programmer and logic.
There are few assertions that programmer always want to be  True to avoid code failure
This type of requirment is fullfilled by assert statement. This statement takes a
Boolean Condition output of which if True, further program executes. if output of asse
statement is False it raises an Assertion Error.
```

Q5. Why do you no longer use string-based exceptions?

In [ ]:

```
Ans: String-based Exceptions doesn't inherit from Exceptions. so plain exceptions cat
all exceptions and not only system.
```