Assignment 08 Solutions

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

Ans: `raise` and `assert` are the two latest user-defined exception constraints in Python 3.X

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

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.

In [1]:

```
class ToYoungException(Exception):
    def __init__(self,msg):
        self.msg = msg
age = 14
if age <= 18: raise ToYoungException('To Young For Voting')
```

```
---------------------------------------------------------------------------
ToYoungException                          Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_5132/1974390838.py in <module>
      3         self.msg = msg
      4 age = 14
----> 5 if age <= 18: raise ToYoungException('To Young For Voting')

ToYoungException: To Young For Voting
```

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

Ans: The `process()` method of LoggerAdapter is where the contextual information is added 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 ?

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 assert statement is False it raises an `Assertion Error` .

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

Ans: Accordig to recent updates the standard exceptions are Python classes, and a few new standard exceptions have been added. The obsolete AccessError exception has been deleted. Because it is possible (although unlikely) that this change broke existing code, the Python interpreter can be invoked the command line option -X to disable this feature, and use string exceptions like before. This option is a temporary measure - eventually the string-based standard exceptions will be removed from the language altogether. It hasn't been decided whether user-defined string exceptions will be allowed in Python 2.0. String-based Exceptions doesn't inherit from Exceptions. so plain exceptions catch all exceptions and not only system.