

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

Answer: There are multiple latest exception constraints in python 3.10

- 1) multiple exception types must be parenthesized
- 2) try blocks without except or finally blocks will throw the error “expected expect or finally block”
- 3) [SyntaxError](#) exceptions now have end_lineno and end_offset attributes. They will be None if not determined
- 4) Literal objects will now raise a [TypeError](#) exception during equality comparisons if any of their parameters are not [hashable](#). Note that declaring Literal with unhashable parameters will not throw an error:

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

Answer: Try and except statements are used to catch and handle exceptions in Python. Statements that can raise exceptions are kept inside the try clause and the statements that handle the exception are written inside the except clause.

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

Answer: 1) the first method is using “<Exception Object>.args” The first argument in args is the message.

Example :-

except Exception as ex:

Ex.args = (“message”,)

2) When raising a new exception while another exception is already being handled, the new exception’s `__context__` attribute is automatically set to the handled exception. This implicit exception context can be supplemented with an explicit cause by using `from with` [raise](#)

Example :-

raise new_exc from original_exc

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

Answer: 1) Method one is capturing the Exception attributes into an object

except Exception as e:

logger.error('Failed to upload to ftp: '+ str(e))

2) Method two is the older way, which is as follows :-

```
try:
    ... some operation(s) ...
except:
    exc_type, exc_value, exc_traceback = sys.exc_info()
    ... exception handling ...
```

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

Answer: Because in Python 3, Exceptions are modelled as instances of dedicated classes, which contains all information needed about the error, its type, value and traceback