1. 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

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

Ans: 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.

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

Ans: process() method: contextual information is added to the logging output. it passes the message and keyword arguments of the logging call, and it passes back modified versions to be used in the call to the underlying logger.
exception() method: Logs a message with level ERROR on this logger.

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

Ans: *Assert* and *raise* are two methods for specifying the text of an exception object's error message.
raise triggers an explicit exception, if a certain condition is True
assert takes a boolean condition output and executes program if True and raises an Assertion Error if False.

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

Ans: String-based Exceptions don't inherit from Exceptions.