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

In Python 3.8, the except statement was updated to allow for multiple exception types to be handled in a single block. This can be done by using a comma-separated list of exception types in the except clause.

Python 3.8 also introduced a new exception type called StopIteration. This exception is raised when the iter() function reaches the end of an iterable object. This can be useful for breaking out of loops when it is known that the iterable object is empty.

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

class-based exceptions are matched to handlers based on the class hierarchy of the exception classes, starting from the point where the exception was raised. The first handler that is found in the class hierarchy that is compatible with the exception object is the one that will be executed.

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

1. Using the with statement:

The with statement will automatically close the file my\_file.txt when it is done being used, even if an exception is raised. This ensures that the file is closed properly and that any data that has been written to it is not lost.

1. Using the contextlib.contextmanager decorator:

The contextlib.contextmanager decorator can be used to create a context manager that automatically performs some cleanup action, such as closing a file, when it is done being used. This is useful for ensuring that resources are properly cleaned up even if an exception is raised.

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

* Using the str() method: The str() method returns a string representation of the exception object, including the error message. For example, the following code will print the error message of an AttributeError exception:
* Using the args attribute: The args attribute of an exception object is a tuple of values that are passed to the exception when it is raised. The first value in the args tuple is the error message. For example, the following code will print the error message of an AttributeError exception:

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

1. String-based exceptions are less readable and maintainable.
2. String-based exceptions are less efficient.
3. String-based exceptions are less flexible.