Q1. What 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?

Class-based exceptions can be declared in the interface of procedures. ... In Class Builder and Function Builder, this done by selecting exception classes when defining exceptions in the interface. The declared exceptions can occur at the call position of a procedure if the exception is not handled in the procedure.

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

Using arguments for Exceptions in Python is useful for the following reasons:

* It can be used to gain additional information about the error encountered.
* As contents of an Argument can vary depending upon different types of Exceptions in Python, Variables can be supplied to the Exceptions to capture the essence of the encountered errors. Same error can occur of different causes, Arguments helps us identify the specific cause for an error using the except clause.
* There are a similar set of functions for performing the same operations with the current call stack instead of a traceback.

### print\_stack()

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

If you are going to print the exception, it is better to use print(repr(e)); the base Exception.\_\_str\_\_ implementation only returns the exception message, not the type. Or, use the traceback module, which has methods for printing the current exception, formatted, or the full traceback

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

In Python versions 1.5 and later, 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.