1.

Python also provides an exception handling method with the help of try-except. Some of the standard exceptions which are most frequent include IndexError, ImportError, IOError, ZeroDivisionError, TypeError, and FileNotFoundError.

2.

Class-based exceptions are realized as instances of exception classes. Exception classes are either predefined globally in the system or can be defined by the user (globally or locally). Class-based exceptions are raised either by the ABAP runtime environment or by a program.

3.

Python has two types of exceptions namely, Built-In Exceptions and User-Defined Exceptions

A better strategy is to just append your message to the argument of the original exception if possible as in err. args += ("message",) and re-raise the exception message. The traceback might not take you to the line numbers where the exception was caught but it will take you to where the exception occurred for sure.

4.

Use except Exception as to catch an exception and save its error message. Place the code where the exception may occur in a try block. Immediately after the try block, make an except block with except Exception as e to handle any exceptions that occur where e is the exception object, which can be printed.

5.

Python's standard exceptions (IOError, TypeError, etc.) were defined as strings. ... 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.