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

Answer:raise and assert 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?

Answer: Class-based exceptions are raised either by the ABAP runtime environment or by a program. Exception situations recognized by the system, and whose causes can be handled in the program, raise predefined class-based exceptions.Raising an exception is a technique for interrupting the normal flow of execution in a program, signaling that some exceptional circumstance has arisen, and returning directly to an enclosing part of the program that was designated to react to that circumstance.

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

Ansswer: There might arise a situation where there is a need for additional information from an exception raised by Python. Python has two types of exceptions namely, Built-In Exceptions and User-Defined Exceptions. Why use Argument in Exceptions? 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.

It can also be used to trap multiple exceptions, by using a variable to follow the tuple of Exceptions.

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

Answer: The parser repeats the offending line and displays a little ‘arrow’ pointing at the earliest point in the line where the error was detected. The error is caused by (or at least detected at) the token preceding the arrow: in the example, the error is detected at the function print(), since a colon (':') is missing before it. File name and line number are printed so you know where to look in case the input came from a script.

msgText = getReport(exception) gets the error message for an exception and returns it as formatted text, msgText . The message is the value of the message property of the MException object, exception . It is the same text that MATLAB® displays when it throws the exception.

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

Answer: String-based Exceptions doesn't inherit from Exceptions. so plain exceptions catch all exceptions and not only system.