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

We can define the user defined exception constraints with the help of assert and raise keywords and handling the exceptions in the try – except block.

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

The exceptions which are raised need to define in the try and except block to get this handled.

If there is a specific except block is present in the try block and it’s matching with the raised exception then that exception block will get executed. Or else if there is no match then super class of all exception will handle the respective exception.

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

We can add the messages in the exception block by printing the e.message from the exception object.

OR we can give the custom message / comment in the exception block to give more details on to it.

Also we can add more details by putting the raise statement and add the detailed information into it by mentioning the error.

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

By printing the exception e we can get the text of the exception like below:

Except Exception as e:

Print(e)

Or by printing the variable itself from the try block which causing the exception.

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

There are already built in exception has been used for the standard exception and also we can get the details by printing the exception as stated in the question 4 so there is no need to use the string based exceptions.