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

Ans: **we have to create a class that implements the Exception class and then we have to use the raise statement with the user defined class name. Assert statement is also used for user defined exception.**

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

Ans: **whatever exception is raise by the try block goes to the except block where the message thrown by the try block gets matched if it matches with the class used in the except block it is handled but if its not then it would still go unhandled despite using the except block due to incorrect class in the block being used.**

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

Ans: **presuming ‘context’ word here is showing the error message for which the error has occurred.**

**So to show the context of the error information either we can use the base class ‘Exception’ or we can use a specific calss like ‘IO or AssertionError or NameError’**

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

Ans: **You can use the raise statement and then add a message or you can create a custom class and then add a message in the class itself so when the raise statement creates an object of the custom class it returns the object’s error message.**

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

Ans: **classes can be used for exceptions as they have a lot of nice properties. In the older version there was a use of string based exceptions but not anymore as classes are used for exceptions with more features.**