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

Ans. In python 3.X, exceptions must be defined by classes. Also it should derived from base class Exception

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

Ans. Class-based exceptions match by superclass relationships: naming a superclass in

an exception handler will catch instances of that class, as well as instances of any

of its subclasses lower in the class tree. Because of this, you can think of superclasses

as general exception categories and subclasses as more specific types of exceptions

within those categories.

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

Ans 1.By passing the instance attributes in the object raised, usually in a custom class constructor.

2.Built-in Exception superclass provide a constructor that stores its

arguments on the instance automatically

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

Ans 1.By overloading the \_\_str\_\_ method

2.By using the Exception superclass which will raise the error by default

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

Ans string-based exceptions did not support categories, state information, or behavior inheritance in the way class-based exceptions do so they have been deprecated in latest version of python.