**Assignment 8**

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

Ans: User-defined exceptions are created by inheriting any class by Exception parent class. There are no as such constraints on user-defined exception classes except they must follow the correct python syntax and usage.

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

Ans: When we define a user-defined exception class and if we try to raise this exception class by using the raise keyword, then if in the except clause we specified the previous exception class name or its subclass then the error will be handled by the already defined exception class. If the thrown error does not get matched to any handler, then the system will throw the Exception.

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

Ans: Two methods for attaching context information to the exception are:

1. Using the e.context method: where e is the exception alias name and context is the inbuilt keyword for the exception definition storage.
2. Simply printing the information passed as an argument in the exception class: Suppose we have the following code. It will print whatever we will pass in the anyException argument.

Try:

Raise anyException(“Error!”)

Except anyException as e:

Print(e)

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

Ans: The first method for specifying the text of an exception object’s error message is described in the second part of the above problem I.e., passing the error message in the class argument itself. Second method we can use is by using args keyword this means we can use e.args=("Error!”,) code line to set error message to any desired string. Args keyword is used to assign tuple value to the e variable.

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

Ans: String based exceptions are used in python 2 but not in python 3 because of their many drawbacks. One of them is we cannot inherit string-based exceptions unlike in python 3 class-based exceptions. Due to which exceptions were hard to maintain and were limited. Second major drawback is that string-based exceptions are not modified as per need, whereas we can modify the class-based exceptions by various methods discussed in the above questions. Therefore, due to these reasons string-based exceptions are no longer used in python 3.