**Python advance assignment -8**

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

The two latest user-defined exception constraints in Python 3.X are:

1. In Python 3.x, exceptions are classes rather than strings.
2. In Python 3.x, exceptions are less prone to mistakes due to the built-in exception hierarchy, which makes it easier to catch and handle specific exception types.

**Q2. How are class-based exceptions that have been raised matched to handlers?**  
 When an exception is raised, the interpreter searches the call stack for a matching handler. The match is based on the class of the exception object and the class or superclass specified in the except clause. If a match is found, the corresponding except block is executed and the search for a matching handler stops.

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

1. One method is to include additional information in the exception object, such as an error code or a message describing the context of the error.
2. Another method is to use the **with** statement to attach context-related information, such as the filename or line number, to the exception object.

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

1. One method is to pass a string argument to the exception class's constructor, which is then stored as the **args** attribute of the exception object.
2. Another method is to define a **\_\_str\_\_** or **\_\_repr\_\_** method in the exception class that returns the desired error message.

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

String-based exceptions are less flexible, less powerful and less precise than class-based exceptions. They do not provide any information about the type of error that occurred, making it difficult to write robust and maintainable code. Class-based exceptions can be subclassed and more easily differentiated in handling. They also allow to add more information about the error, like error-code, context etc.