ASSIGNMENT - 6

Q1. Describe three applications for exception processing.

Ans: Applications for exception processing:

* Error Handling: Exceptions are commonly used to handle errors that might occur during code execution, like file I/O errors, division by zero, or incorrect data types.
* Input Validation: Exceptions can validate input data and handle cases where the input doesn't meet the required criteria or format.
* Resource Management: Exceptions assist in managing resources, ensuring that they are properly released or closed even in the presence of errors (e.g., closing files or database connections in a finally block).

Q2. What happens if you don’t do something extra to treat an exception?

Ans: If you don't handle an exception, it will propagate up the call stack until it reaches the top level of the program, resulting in the script terminating abruptly. Python displays the exception traceback, which includes information about the exception and where it occurred.

Q3. What are your options for recovering from an exception in your script?

Ans: Options for recovering from an exception:

* Try-except blocks: Wrap code that might raise an exception within a try block and handle the exception using one or more except blocks. This allows you to execute alternative code or provide a fallback mechanism in case of an exception.
* Finally block: Use a finally block to ensure certain code executes regardless of whether an exception occurs or not. This block is commonly used to perform cleanup operations (e.g., closing files or releasing resources).

Q4. Describe two methods for triggering exceptions in your script.

Ans: Triggering exceptions in your script:

* raise statement: Explicitly raise an exception using the raise statement. You can raise built-in exceptions or create custom exceptions by defining new exception classes.
* Using built-in functions: Certain built-in functions or methods can raise exceptions. For instance, trying to access a non-existent key in a dictionary using dict[key] or attempting to convert an invalid string to an integer using int('abc') will raise KeyError and ValueError respectively.

Q5. Identify two methods for specifying actions to be executed at termination time, regardless of whether or not an exception exists.

Ans: try-except-finally block: The finally block executes cleanup code regardless of whether an exception occurred. This block is executed at the end of a try block, whether an exception was raised or not.

atexit module: The atexit module provides a way to register functions to be executed when the Python interpreter exits. The atexit.register() function can be used to register functions that should run on normal script termination.