Q1. Describe three applications for exception processing.

Ans1

Exception processing is a technique used in programming to handle unexpected or erroneous situations that can arise during the execution of a program.

Error handling in software applications: Exception processing is commonly used to handle errors that can occur in software applications.

File I/O operations: Exception processing is also commonly used in file I/O operations, such as reading and writing files. If there is an error while reading or writing a file, an exception can be thrown and handled using exception processing techniques.

Database operations: Exception processing is also commonly used in database operations.

Q2. What happens if you don&#39;t do something extra to treat an exception?

Ans2

The program will terminate abruptly with an error message or stack trace. This can be disruptive to the user and potentially harmful to the system.

When an exception occurs in a program, it creates a disruption in the normal flow of execution, causing the program to jump to a specific error-handling routine, if one exists. If an error-handling routine is not present, the exception will propagate up the call stack until it reaches the top-level of the program, where it will terminate the program and display an error message or a stack trace.

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

Ans3

Try-Catch blocks: The most common way to recover from an exception is by using a try-catch block. The code that might throw an exception is placed within a try block, and any exceptions that are thrown are caught and handled within the catch block.

Raising Exceptions: In some cases, you might want to raise an exception intentionally to signal a specific error condition.

Logging: Logging is another way to handle exceptions in a script.

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

Ans4

Using built-in Exception classes: Several built-in exception classes that can be raised to trigger exceptions intentionally.

Using assert statements: Assert statements are used to test a condition and trigger an AssertionError exception if the condition is not met.

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

Ans5

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

The try-finally block: The try-finally block ensures that a particular block of code is executed before the program exits, regardless of whether or not an exception is raised. The code in the finally block is executed regardless of whether or not an exception is raised.

The atexit module: The atexit module provides a way to register functions to be executed when the Python interpreter terminates, regardless of whether or not an exception is raised. This can be useful for cleaning up resources or performing final tasks before the program exits.