Q1. Describe three applications for exception processing.

Errors that occur at runtime (after passing the syntax test) are called exceptions or logical errors.

For instance, they occur when we try to open a file(for reading) that does not exist (FileNotFoundError), try to divide a number by zero (ZeroDivisionError), or try to import a module that does not exist (ImportError).

Whenever these types of runtime errors occur, Python creates an exception object. If not handled properly, it prints a traceback to that error along with some details about why that error occurred.

Illegal operations can raise exceptions. There are plenty of built-in exceptions in Python that are raised when corresponding errors occur.

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

An exception object is created when a Python script raises an exception. If the script explicitly doesn't handle the exception, the program will be forced to terminate abruptly

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

In Python, exceptions can be handled using a try statement.

The critical operation which can raise an exception is placed inside the try clause. The code that handles the exceptions is written in the except clause.

we did not mention any specific exception in the except clause.

 it will catch all exceptions and handle every case in the same way. We can specify which exceptions an except clause should catch.

A try clause can have any number of except clauses to handle different exceptions, however, only one will be executed in case an exception occurs.

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

raise

raise statement without any arguments re-raises the last exception. This is useful if you need to perform some actions after catching the exception and then want to re-raise it. But if there was no exception before, raise statement raises TypeError Exception.

raise exception (args) from original\_exception

This statement is used to create exception chaining in which an exception that is raised in response to another exception can contain the details of the original exception - as shown in the example below.

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

Finally block always executes irrespective of an exception being thrown or not. The final keyword allows you to create a block of code that follows a try-catch block.

Finally, clause is optional. It is intended to define clean-up actions which should be that executed in all conditions.