Q1. Describe three applications for exception processing.

Sol: Error Handling in software development: Exception processing is commonly used in software development to handle errors and unexpected situations.

Financial Transactions: Exception processing is also used in financial transactions to handle situations where there are errors or inconsistencies in the data being processed.

Telecommunications network: Exception processing is also important in telecommunication networks, where network failures and other errors can occur frequently

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

Sol: if you don't do something extra to treat an exception, the program may terminate abruptly or behave unexpectedly.

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

Sol: 1. Ignore the exception

2. Retry the operation

3. Handle the Exception

4. Rollback the transaction

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

Sol: 1. Raise an Exception: You can explicitly raise an exception in your script by using the “raise” statement.

2. Use built-in Exceptions: Python provides a number of built-in exceptions that can be raised to indicate specific error conditions.

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

Sol: 1. The try-finally Block: The try-finally block is a construct that allows you to specify cleanup code that should be executed regardless of whether or not an exception is thrown

2. The with Statement.