Q1. Describe three applications for exception processing.

Python provides the number of built-in exceptions, but here we are describing the common standard exceptions. A list of common exceptions that can be thrown from a standard Python program is given below.

1. **ZeroDivisionError**: Occurs when a number is divided by zero.
2. **NameError**: It occurs when a name is not found. It may be local or global.
3. **IndentationError**: If incorrect indentation is given.
4. **IOError**: It occurs when Input Output operation fails.

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

When an exception occurs, if you don't handle it, the program terminates abruptly and the code past the line that caused the exception will not get executed.

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

* A single try statement can have multiple except statements. This is useful when the try block contains statements that may throw different types of exceptions.
* You can also provide a generic except clause, which handles any exception.
* After the except clause, you can include an else-clause. The code in the else-block executes if the code in the try: block does not raise an exception.
* The else-block is a good place for code that does not need the try: block's protection.

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

1. Try – This method catches the exceptions raised by the program.
2. Raise – Triggers an exception manually using custom exceptions.

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