Q1. Describe three applications for exception processing.

Ans. Exception Processing is important to find exceptions that causes the runtime error. As runtime errors Halt the program execution when exception occurs.

Exception Processing is used in Various Applications of which few examples are:

1. Checking Appropriate use of input in an application

2. Checking for Arithmetic exceptions in mathematical executions

3. Checking File I/O exceptions during File handling

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

Ans. When an exception occurred, 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?

Ans. Python provides try and except statements for recovering from an exception in your script.

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

Ans. To avoid exception in a code or script, there are two methods to handle Python exceptions:

Try – This method catches the exceptions raised by the program.

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.

Ans. Python Provides else and finally blocks for specifying actions to be executed at termination time, regardless of whether an exceptions exists or not.