

**Q1. Describe three applications for exception processing.**

Answer: Exception Processing is important to find exceptions that cause runtime error, that causes halting of program execution at that instance when exception occurs. Exception processing is used in various applications of which few examples are checking appropriateness of use input in an application, checking for arithmetic exceptions in mathematical executions, checking of file I/O exceptions during file handling.

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

Answer: If exception are not handled flow of program is broken if an exception is being raised during running a program, inability of program to handle this exception will result in crashing of program flow.

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

Answer :Use of try and except block can be used for recovering from an exception in script.

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

Answer: raise and assert are the two methods that can be used to trigger exceptions in python.

raise method triggers an exception if condition provided by raise turns out to be true.

E.g.

```
x = 10
```

```
if x > 5:
```

```
    raise Exception('x should not exceed 5. The value of x was: {}'.format(x))
```

assert let the program to execute if certain condition is met i.e. is True, else exception is raised.

E.g.

```
import sys
```

```
assert ('linux' in sys.platform), "This code runs on Linux only."
```

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

Answer: else and finally are two methods for specifying actions to be executed at termination time, regardless of whether or not an exception exists.