Q1. Describe three applications for exception processing.

try:  
 f = open('sudh.txt', 'r')  
except Exception as e:  
 print(e)  
else:  
 print("File opened successful")  
 f.close()  
  
print('Program is not interrupted')

[Errno 2] No such file or directory: 'sudh.txt'

Program is not interrupted

try:  
 raise NameError('sudh')  
except NameError as e:  
 print("Name error")

Name error

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

Whenever an exception occurs, the program stops the execution, and thus the further code is not executed. Therefore, an exception is the run-time errors that are unable to handle to Python script. An exception is a Python object that represents an error

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

def ccp():  
 while True:  
 try:  
 a = int(input('Enter an int value::'))  
 except Exception as e:  
 print(e)  
 print('Please enter again an int value')  
 continue  
 else:  
 print(f'Yes you have entered an int {a}')  
 break  
 finally:  
 print('It always execute!!')  
  
print(ccp())

Enter an int value::20

Yes you have entered an int 20

It always execute!!

None

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

Try statement – This method throws the exceptions when the code within try is executed.

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.

* In else block, code is written to handle if no exception is raised by code in try block.
* In finally block, code executes regardless of whether or not an exception exists.