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

**ANS:-**

**Exceptions are raised when the program is syntactically correct, but the code resulted in an error. This error does not stop the execution of the program, however, it changes the normal flow of the program.**

**def div(a,b):**

**try:**

***# code executes here***

**c = a/b**

**print("a/b = ",c)**

***# Using exception object with the except statement***

**except Exception as e:**

**print(e)**

**return "can't divide by zero"**

**else:**

**return "This is else block and executed when code in try excetues without any exception"**

**a = int(input("Enter a:"))**

**b = int(input("Enter b:"))**

**print(div(a,b))**

**Enter a:3**

**Enter b:0**

**division by zero**

**can't divide by zero**

**try:**

**f = open('sample.txt','r')**

**except Exception as e:**

**print(e)**

**else:**

**print("File opened successfully")**

**f.close()**

**print('Program is not interrupted')**

***# If we dont use try except else blocks, normal execution of***

***# program will be interrupted.***

***# with try except, exception is caught when file is not found # but program continues execution***

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

**Program is not interrupted**

**try:**

**raise NameError('akash')**

**except NameError as e:**

**print("Excepton!!")**

**Excepton!!**

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

**ANS:-**

**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?**

**ANS:-**

**In except block, we write a code to handle exception and next set of actions.**

**def askint():**

**while True:**

**try:**

**a = int(input('Enter an integer value::'))**

**except Exception as e:**

**print(e)**

**print('Please enter an integer value')**

**continue**

**else:**

**print(f'Yes you have entered an ineger {a}')**

**break**

**finally:**

**print('Yayy i always executes!!')**

**askint()**

**Enter an integer value::as**

**invalid literal for int() with base 10: 'as'**

**Please enter an integer value**

**Yayy i always executes!!**

**Enter an integer value::6**

**Yes you have entered an ineger 6**

**Yayy i always executes!!**

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

**ANS:-**

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

**Raise – Triggers an exception manually using custom exceptions.**

**Examples are already explained above.**

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

**ANS:-**

**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.**