**FSDS MAY BATCH 2022(Python Assignment -6)**

**Submitted by: Shubham Tiwari**

Q1: Describe three applications for exception processing.

Ans: The three applications can be:

(i) ATM withdrawal of more than the available amount.

(ii)Division by zero.

(iii)Accessing a file which does not exist.

Q2: What happens if you don’t do something extra to treat an exception?

Ans: If we don’t do something to treat an exception it will obviously crash the program or it will force the program to terminate immediately.

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

Ans: Basically exceptions occur 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,so the most probable option is to use “**Try and Except statements**” since these statements are used to catch and handle exceptions in python.

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

Ans: The two methods for handling exceptions in our script can be:

1)Try statement.

Try and except statements are used to catch and handle exceptions in Python. Statements that can raise exceptions are kept inside the try clause and the statements that handle the exception are written inside except clause.This method catches the exceptions raised by the program.

For example: We are considering an array whose index is out of bond and handle the given exception.

a **=** [1, 2, 3]

**try**:

**print** ("Second element = %d" **%**(a[1]))

    # Throws error since there are only 3 elements in array

**print** ("Fourth element = %d" **%**(a[3]))

**except**:

    print ("An error occurred")

**Output:**

Second element = 2

An error occurred

2)Raise statement.

Triggers an exception manually using custom exceptions. The raise statement specifies an argument which initializes the exception object. Here, a comma follows the exception name, and argument or tuple of the argument that follows the comma. The syntax is as follows:

raise [Exception [, args [, traceback]]]

In this syntax, the argument is optional, and at the time of execution, the exception argument value is always none.

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

Ans: The two methods are :

1)**Finally statement.**

Finally statement block always executes irrespective of an exception being thrown or not. The final keyword allows you to create a block of code that follows a try-catch block.

Finally, clause is optional. It is intended to define clean-up actions which should be that executed in all conditions.

For example:

try:

raise KeyboardInterrupt

finally:

print 'welcome, world!'

**Output:**

Welcome, world!

KeyboardInterrupt

2)**Raise statement**

The raise statement specifies an argument which initializes the exception object. Here, a comma follows the exception name, and argument or tuple of the argument that follows the comma.The basic syntax as follows:

raise [Exception [, args [, traceback]]]

In this syntax, the argument is optional, and at the time of execution, the exception argument value is always none.