Python Fundamentals day 63

Today's Agenda

- Exception handling contd

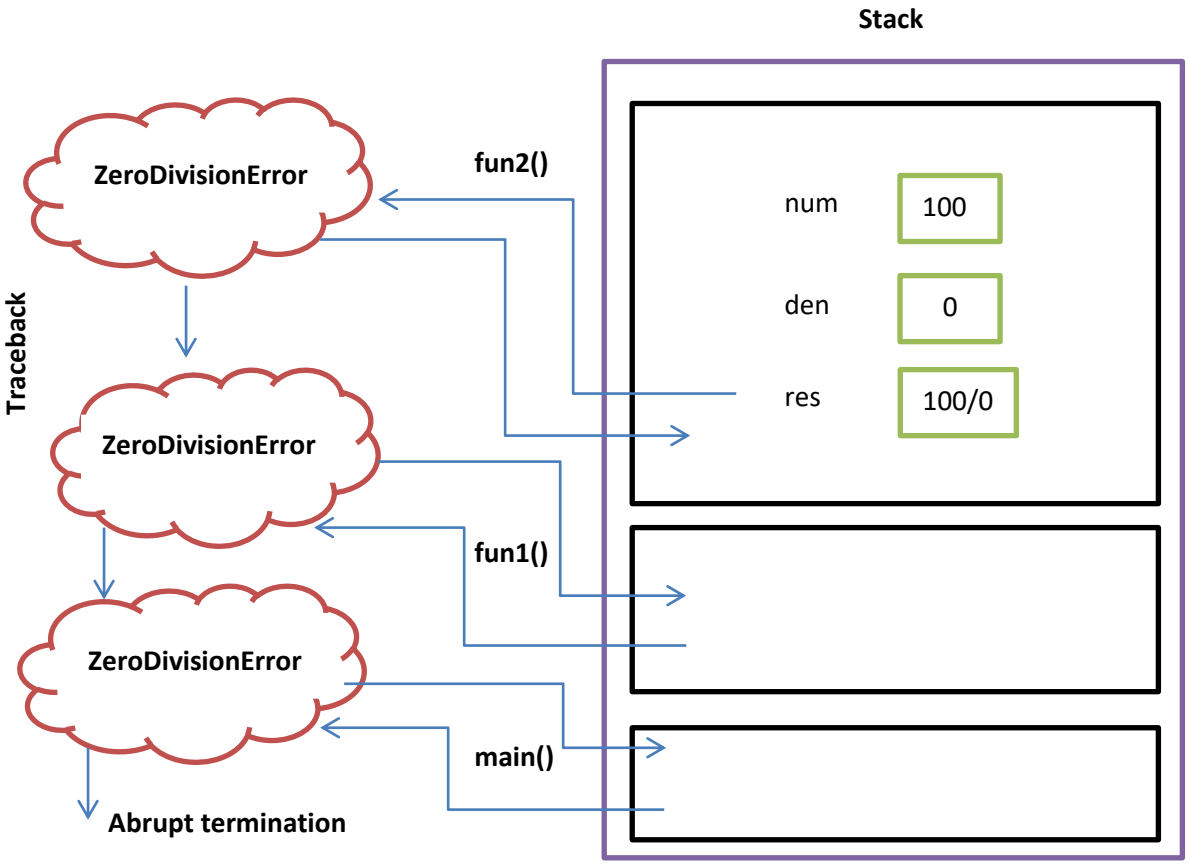
Exception handling contd

Let us see how exception handling mechanism works when multiple method calls are involved through the following example

```
def fun2():
    print('fun2() started execution')
    num=int(input('Enter the numerator'))
    den=int(input('Enter the denominator'))
    res=num/den
    print(res)
    print('fun2() finished execution normally')

def fun1():
    print('fun1() started execution')
    fun2()
    print('fun1() finished execution normally')

def main():
    print('main() started execution')
    fun1()
    print('main() finished execution normally')

main()
```

Output:

```
  File "C:\Users\rooman\Anaconda3\lib\site-packages
\spyder_kernels\customize\spydercustomize.py", line 110,
in execfile
    exec(compile(f.read(), filename, 'exec'), namespace)

  File "C:/Users/rooman/Downloads/test.py", line 19, in
<module>
    main()

  File "C:/Users/rooman/Downloads/test.py", line 16, in
main
    fun1()

  File "C:/Users/rooman/Downloads/test.py", line 11, in
fun1
    fun2()

  File "C:/Users/rooman/Downloads/test.py", line 5, in
fun2
    res=num/den

ZeroDivisionError: division by zero
```



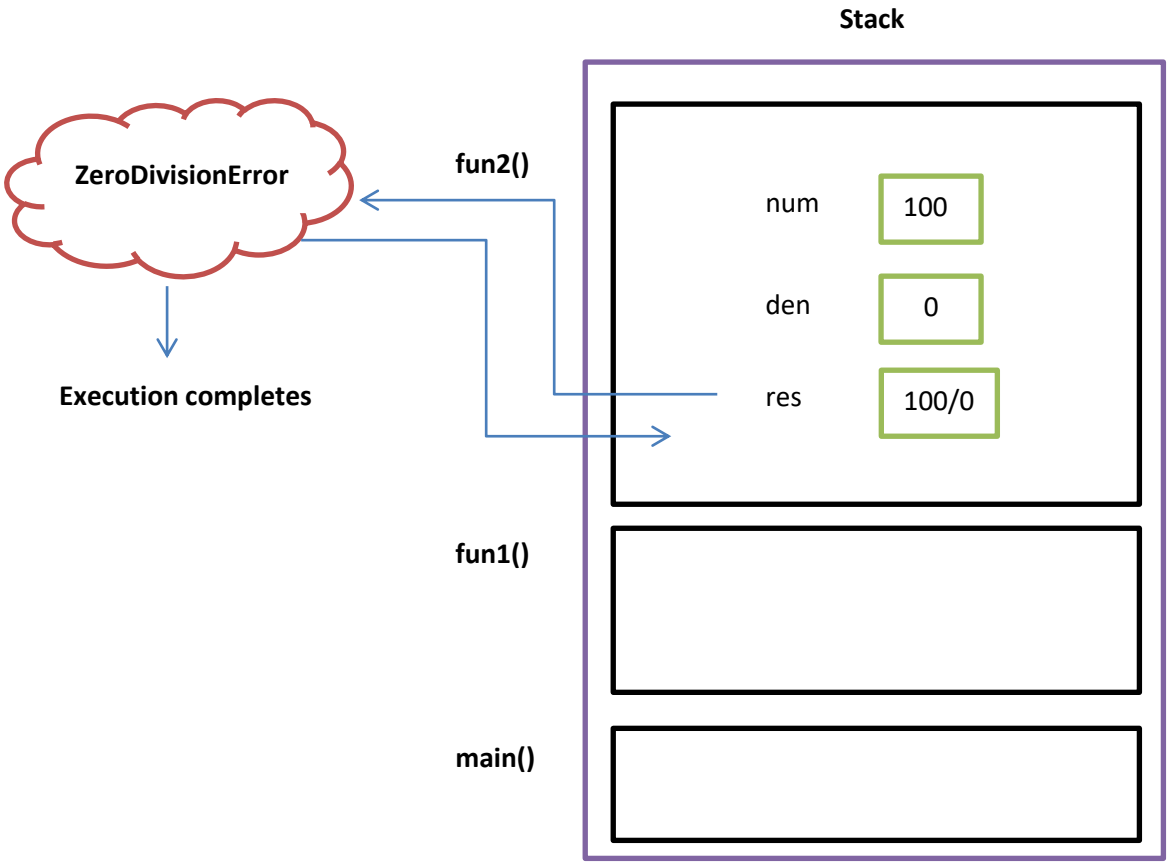
If except block is not in the method where exception is generated then it traces back to the method who called for it and checks in that method. If anywhere it is not present then the program is abruptly terminated.

Let us try to handle this exception

```
def fun2():
    print('fun2() started execution')
    try:
        num=int(input('Enter the numerator'))
        den=int(input('Enter the denominator'))
        res=num/den
        print(res)
    except ZeroDivisionError:
        print('Exception handled in fun2()')

    print('fun2() finished execution normally')

def fun1():
    print('fun1() started execution')
    fun2()
    print('fun1() finished execution normally')

def main():
    print('main() started execution')
    fun1()
    print('main() finished execution normally')

main()
```

Output:

```
In [5]: runfile('C:/Users/rooman/Downloads/test.py',
wdir='C:/Users/rooman/Downloads')
main() started execution
fun1() started execution
fun2() started execution

Enter the numerator100

Enter the denominator0
Exception handled in fun2()
fun2() finished execution normally
fun1() finished execution normally
main() finished execution normally
```

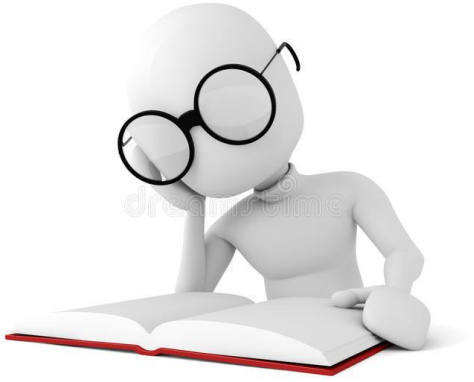


Let us see what happens when the except block is in fun1()

```
def fun2():
    print('fun2() started execution')
    num=int(input('Enter the numerator'))
    den=int(input('Enter the denominator'))
    res=num/den
    print(res)
    print('fun2() finished execution normally')

def fun1():
    print('fun1() started execution')
    try:
        fun2()
    except ZeroDivisionError:
        print('Exception handled in fun1()')

    print('fun1() finished execution normally')

def main():
    print('main() started execution')
    fun1()
    print('main() finished execution normally')

main()
```

Output:

```
In [5]: runfile('C:/Users/rooman/Downloads/test.py',
wdir='C:/Users/rooman/Downloads')
main() started execution
fun1() started execution
fun2() started execution

Enter the numerator100

Enter the denominator0
Exception handled in fun2()
fun2() finished execution normally
fun1() finished execution normally
main() finished execution normally
```



We can see that exception is been handled, but fun2() did not finish it's execution as the control came to fun1() along with exception object.

Let us also try placing the except block in main()

```
def fun2():
    print('fun2() started execution')
    num=int(input('Enter the numerator'))
    den=int(input('Enter the denominator'))
    res=num/den
    print(res)
    print('fun2() finished execution normally')

def fun1():
    print('fun1() started execution')
    fun2()
    print('fun1() finished execution normally')

def main():
    print('main() started execution')
    try:
        fun1()
    except ZeroDivisionError:
        print('Exception handled in main()')

    print('main() finished execution normally')

main()
```

Output:

```
In [7]: runfile('C:/Users/rooman/Downloads/test.py',
wdir='C:/Users/rooman/Downloads')
main() started execution
fun1() started execution
fun2() started execution

Enter the numerator100

Enter the denominator0
Exception handled in main()
main() finished execution normally
```


As we see exception is handled but fun1() and fun2() did not finish their execution.