



Experiment 5

Title : Program to implement exception handling.

Aim : Write a various types of exceptions in python & implement it.

Theory : Exception

An exception is an event, which occurs during the execution of program that disrupts the normal flow of the program is instructions.

When a python script encounters a situation that it cannot cope with, it raises an exception. An exception is a python object that represents an error.

When a python script raises an exception, it must either handle the exception immediately otherwise it terminates & quits.

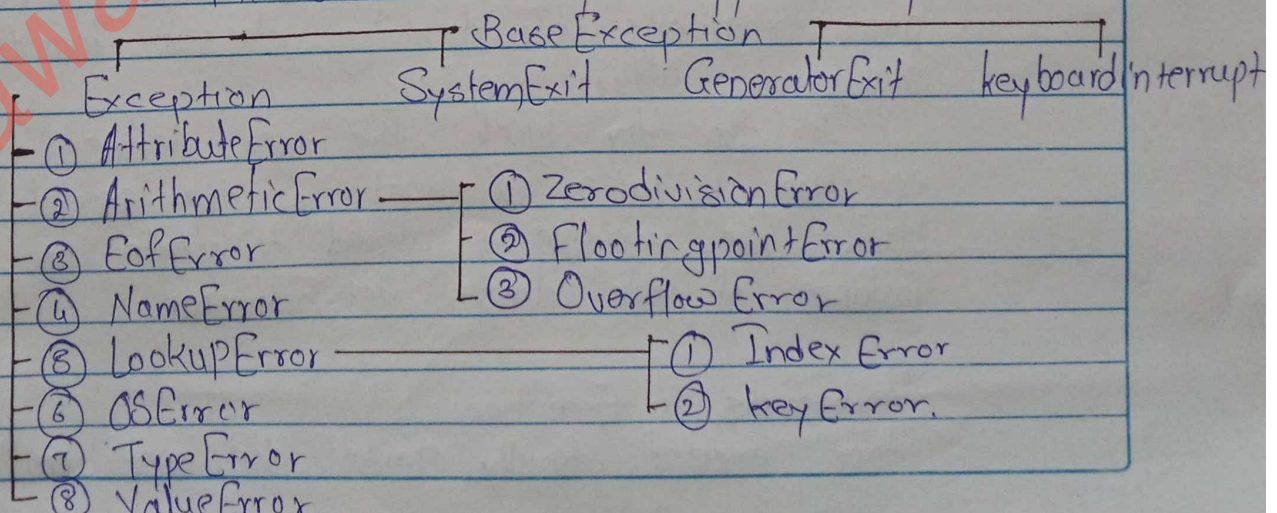
Types of Exception:

1] Built in Exception

They are following some built in exception python are:

AttributeError, ArithmeticError, TypeError, etc.

The basic hierarchy for python Exception classes:





- 1) Exception : Base class for all exceptions.
- 2) SystemExit : Raised by the `sys.exit()` function.
- 3) KeyboardInterrupt : Raised when the user interrupts program execution, usually by pressing `ctrl + c`.
- 4) ArithmeticError : Base class for all errors that occur for numeric calculation.
- 5) AttributeError : Raised in case of failure of attribute reference or assignment.
- 6) EOFError : Raised when there is no input from either the `raw_input()` or `input()` function & the EOF file is reached.
- 7) LookupError : Base class for all lookup errors.
- 8) TypeError : Raised when an operation or function is attempted that is invalid for the specified data type.
- 9) ValueError : Raised when built-in function for a data type has the valid type of args but the args have invalid values specified.

2] User-defined exception :

Python also allows you to create your own exceptions by deriving classes from the standard builtin exceptions.

In the `try` block, the user defined exception is raised & caught in the `except` block. The variable is used to create an instance of the classes `networkerror`.

Keywords in Exception

`try`, `except`, `finally`, `raise`, `else`, `assertion`.



1) try - except

Try & except statements are used to handle errors within our code in python.

try block is used to check some code for errors e.e. the code inside the try block will execute when there is no error in the program the code inside the try block generate error then the it goes to the except for handle this error.

If there is no exception occurs, then only try block execute.

2) finally

Finally block along with a try block, the finally block is a place to put any code that must execute, whether the try-block raised an exception or not.

Finally block is always executed after leaving the try statement.

3) raise

Python raise keyword is used to raise exceptions or errors. the raise keyword raise an error & stops the control flow of the program

4) else

This keyword is used to conjunction the try-except block. The code within the else block is executed if no exception occurs in the try block.

5) assert : This keyword is used to test if a given condition is true, & if not raises an AssertionError exception.



Syntax :

try :

code that might raise an exception

except :

code to handle exception

else :

code to execute if no exception occurs

finally :

code that will be executed regardless of exception

- Example :

try :

num = int(input("Numerator : "))

den = int(input("Denominator : "))

res = num / den

assert den != 0, "Denominator can't Zero"

except ValueError :

print("Error : Invalid Input")

except ZeroDivisionError :

print("Can not divided by zero")

except AssertionError as ex :

print("Assertion error", str(ex))

else :

print("Result : ", result)

finally :

print("End of the program")

Conclusion : Exception handling in python allows for graceful handling of errors & exceptional situations, ensuring that code execution can continue smoothly even in the presence of exceptions.