Exception Handling

in generally any programming language we will get two type's of errors,they are

1).Syntax/compile time errors

2).Runtime errors

Syntax Errors:

the error which occur because of syntax mistakes,that type of errors are called Syntax errors.

(or)

Before starrting the program execution we will get the errors,that type of errors are called synatx errors.

if any syntax error occured in our python program in that case our python program execution will not be started.

ex1:

```
print("hai)
```

syntax: unterminated string literal

sol:

```
print("hai")
```

ex2:

```
print("hai"
```

syntax error: '(' was never closed

sol:

```
print("hai")
```

ex3:

```
print("hai")
if 3<5:
print("siva")
print("bye")
```

syntax error: expected an indentation block after 'if'

```
sol:
---
print("hai")
if 3<5:
    print("siva")
print("bye")

ex4:
----
print("hai")
if 3<5
    print("siva")
print("bye")

syntax error: ':' expected end of if statement

sol:
----
print("hai")
if 3<5:
    print("siva")
print("bye")


Runtime Error's:
---------------
        after starting the program execution we will get the errors,that type of
errors are called Runtime Errors.

        the runtime errors are technically we are calling as Exceptions.

ex:
--
print("hai")
print("siva")
num=int(input("enter your number: "))
if num>5:
    print("good morning")
print("bye")

output:
------
hai
siva
enter your number: 2.3

ValueError: invalid literal for int() with base 10: '2.3'


if any runtime error occured in our python program,in that case our python program
```

```
execution will be terminated abnormally.

what is abnormal termination?
-----------------------------
        the program execution will be terminates at the time of middle of the
program execution,is known as a abnormal termination.


ex1:
---
print("hai")
print("siva")
num=int(input("enter your number: "))
if num>5:
    print("good morning")
print("bye")

output: normal termination
------
hai
siva
enter your number: 9
good morning
bye

output2: abnormal termonation
-------
hai
siva
enter your number: 2.3
ValueError: invalid literal for int() with base 10: '2.3'

what are the reasons to get the runtime errors?
-----------------------------------------------
        1).invalid data

        2).invalid logic

        3).index out of range

        4).missing key's

        5).memory issues

                .....
                .....

types of exceptions:
-------------------
the exception's can be categorized into two type's they are
```

        1).Builtin/predefined exceptions

        2).user-defined exceptions

Builtin/predefined exceptions
----

        these exception's are already predefined for some special purpose,we can use that exception's only for that particular reason's/situation's only.

        every builtin exception haveing it's own exception class,the builtin exception's are automatically raised but we can handle the builtin exception's manually.

what is Exception Handling?
----

        whenever Exception will be occured in our programm,then immediately to identify which exception is occured,to create that exception class object(raise the exception),receve that exception class object and forwarding it.

        in python we can handle the exceptions by using try,except and finally blocks.

try block:
----

we can define any block,which is preceeded by try keyword that block is called try block.

```
        try:
           stmt_1
           stmt_2
           .....
           stmt_n
```

which statements causes to runtime errors and which statements execution is depends on runtime error occured statements,that type of statements we are represented in try block.

if any exception occured in our try block,then immediately our try block to identify which exception will be occured,to rasie that exception(to create that exception class object),Receiveing that exception class object and forwarding to the except block.

except block
----

        we can define any block,which is preceeded by 'except' keyword that block is called except block.

        in except block we can write the user friendly messages to the programmer/developer/user related to that particular exceptions.

```
        the except block can be categorized into two type's they are

        1).default except block

        2).Named except block.

default except block:
--------------------
        the default except block can handle any type of exceptions.

        except:
              stmt_1
              stmt_2
              ......
              stmt_n

ex:
---
print("hai")
print("siva")
try:
    num=int(input("enter your number: "))
    if num>5:
        print("good morning")
except:
    print("enter integers only")
print("bye")

output: without any exception
------
hai
siva
enter your number: 7
good morning
bye

output2: with exception can be handled
-------
hai
siva
enter your number: 2.3
enter integers only
bye


ex3:
---
print("hai")
try:
    x=int(input("enter x value: "))
    y=int(input("enter y value: "))
    z=x/y
    print(z)
except:
    print("enter integers only")
print("bye")
```

```
output1: without any exception
-------
hai
enter x value: 3
enter y value: 2
1.5
bye
```

```
output2: with exception can be handled
-------
hai
enter x value: 3
enter y value: 2.3
enter integers only
bye
```

```
output3: with exception canbe handled
-------
hai
enter x value: 3
enter y value: 0
enter integers only
bye
```

Named except block
----
        the named except block can be handled that particular type of exception's only.

```
        except ExceptionClassName:
                stmt_1
                stmt_2
                .....
                stmt_n
```

ex:

```
print("hai")
try:
    x=int(input("enter x value: "))
    y=int(input("enter y value: "))
    z=x/y
    print(z)
except ValueError:
    print("enter integers only")
print("bye")

output: without any exception
------
hai
enter x value: 3
enter y value: 2
1.5
bye

output2: with exception can be handled
-------
hai
enter x value: 3
enter y value: 2.3
enter integers only
bye

output3: with exception can't be handled
--------
hai
enter x value: 3
enter y value: 0
ZeroDivisionError: division by zero

ex2:
---
print("hai")
try:
    x=int(input("enter x value: "))
    y=int(input("enter y value: "))
    z=x/y
    print(z)
except Exception as e:
    print(e)
print("bye")

output:
------
hai
enter x value: 3
enter y value: 2
1.5
bye

output2:
-------
```

```
hai
enter x value: 3
enter y value: 0
division by zero
bye

output3:
-------
hai
enter x value: 2
enter y value: 3.2
invalid literal for int() with base 10: '3.2'
bye
```

note:
----
if any exception occured in our try block then immediately control will come out from try block,in that case dont execute the remaing statement's in our try block.