Errors and Exception Handling in Python

Types of errors

- Compile time error – errors that occur during compilation.
  - Syntax errors occurs when rules of the language are not followed

    eg: **If** a>b (if is the keyword)
  - Semantic – meaningless statement

    eg: a+b = c (semantically wrong)
- Runtime errors- Errors that occur during execution of the program which is hard to detect

  eg: divide by zero error
- Logical errors – When there is some mistake in the logic of the program

  eg: to add two numbers : c= a- b is given

Exceptions in Python

An exception is also an error that occurs during the execution of the program. Whenever there is an error, Python generates an exception that can be handled.

Difference between an error and an exception is that error occurs due to a bug in the program whereas exception occurs out of an unexpected situation while executing the program.

Exception handling

Handling the runtime errors is known as exception handling.

There are lot of errors that occur while writing a python program such as type error, EOFError, FileNotFound error and so on… To handle such errors we have try…except block in Python.

try… except

try block has statements that can generate errors or exceptions

except block has statements that can handle errors.

Example program

```
#Program to divide two numbers
a= int(input("Enter a number"))
b= int(input("Enter another number"))
try:
    c= a/b
    print(c)
except ZeroDivisionError:
    print("Cannot divide a number by zero")
```