Chapter - 1 [Exception Handling in Python]

Syntax Errors:-

These errors are also known as parsing errors.

After syntax error, we have to solve the errors, save and rerun the program.

Exceptions:-

This is shown when code is syntactically correct, there might arise an error during its execution.

For ex:- Trying to open a file that does not exist, division by zero, and so on.

These errors are shown while running the code.

An exception is said to have been raised.

Built-in exceptions:-

Commonly occurring exceptions are usually defined in the compiler/interpreter. These are called built-in exceptions.

Built-in exception ---> Explanations.

SyntaxError --> It is raised when there is an error in the syntax of the Python code.

ValueError --> It is raised when a built-in method or operation receives an argument that has the right data type but mismatched or inappropriate values.

IOError --> It is raised when the file specified in a program statement cannot be opened.

KeyboardInterrupt --> It is raised when the user accidentally hits the Delete or Esc key while executing a program due to which the normal flow of the program is interrupted.

ImportError --> It is raised when the requested module is not found.

EOFError --> It is raised when the end of file condition is reached without reading andy data by input().

ZeroDivisionError --> It is raised when the denominator in a division operation is zero.

IndexError --> It is raised when the index or subscript in a sequence is out of range.

NameError --> It is raised when a local or global variable name is not defined.

IndentationError --> It is raised due to incorrect indentation in the program code.

TypeError --> It is raised when an operator is supplied with a value of incorrect data type.

OverFlowErrror --> It is raised when the result of the calculation exceeds the maximum limit for a numeric data type.

Raise statement:-

It is used to throw an exception that makes by the programmer.

Syntax:-
raise exception_name("Message you want to show")

The argument is generally a string that is displayed when the exception is raised.

whenever the exception will occur the message will display with a brief description is raised.

There are two types of exceptions built-in or user-defined.

The assert statement:-

It is used to test an expression in the program code.

If the result after testing comes false, then the exception is raised.

Process of handling exception:-

When an error occurs, the Python interpreter creates an object called the exception object. This file object contains information about the error like:- what type of error, file name, and line/position where the error has occurred.

The process of creating an exception object and handling it over to the runtime system is throwing an exception.

The runtime system searches the entire program for a block of code, called the exception handler, that can handle the raised exception.

The call stack is the process in which the runtime searches for the exception and exception object.

When the call stack found a suitable handler then it is executed by the runtime process, and this process is known as catching the exception.