

## EXCEPTION HANDLING

1.What is exception? An exception is a type of error that occurs during the execution of a program. An exception is a Python object that represents an error. Python has many builtin exceptions which forces the program to output an error when something in it goes wrong. An exception is the result of an unpredicted event that occurs while your program is running, and prevents the program from doing what it should. As a result, the program throws an exception code.

2. Can you explain the difference between errors and exceptions? An error is an issue in a program that prevents the program from completing its task. In comparison, an exception is a condition that interrupts the normal flow of the program. an error is something that most of the time you cannot handles it. exception are those which can be handled at the run time .

3.How can you ensure that a certain code block runs no matter whether there's an exception or not? Finally clause runs no matter whether there's an exception or not.

4.Can you give me some examples of built-in exceptions? ZeroDivisionError, OverflowError, NameError, TypeError, KeyError, AttributeError.

5.How do you handle exceptions with try/except/finally? The try block lets you test a block of code for errors. The except block lets you handle the error. The finally block lets you execute code, regardless of the result of the try- and except blocks.

6.'Every syntax error is an exception but every exception cannot be a syntax error'. Justify the statement. Syntax errors are detected when we have not followed the rules of the particular programming language while writing a program and that may cause of arising an exception, while exception can occur even without any syntax error For example exception like trying to open a file which does not exist, division by zero, etc. so this is how Every syntax error is an exception but every exception cannot be a syntax error. Exception can be occurred due to Syntax Errors, Logical Error and Run Time Errors, while Syntax errors occurs only due to syntax.

7.What is the use of raise statement? It is used to raise an exception. Raise keyword is used to signal that an error has occured and to specify the type of error.

8.What is the use of else block in exception handling? The else block lets you execute code when there is no error. It is useful for the code that must be executed if the try block does not raise an exception.

9.What is the syntax for try and except block? The try block has any statement to check the occurence of expection. Except block executes if there is any expection occurs in try block.

10.What is the purpose of finally block? The finally block will be executed no matter if the try block raises an error or not. This can be useful to close objects and clean up resources. We generally use the finally block to execute clean up code like closing connections, closing files, or freeing up threads, as it executes regardless of an exception.

11.Can you explain the difference between try-except and try-finally block? The try block contains the code that may raise exceptions or errors. The except block is used to catch the exceptions and handle them. If the try block raises an error, the except block will be executed.

When an exception is thrown in the try block, the execution immediately passes to the finally block

12.Can you give example of using multiple except block in try-except block? Try-except blocks can be used to catch and respond to one or multiple exceptions. In cases where a process raises more than one possible exception, they can all be handled using a single except clause.