Exception Handling

Date	18.11.2022
Team Id	PNT2022TMID34709
Project Name	Personal Assistance For Seniors Who Are Self-Reliant

Exception Handling:-

Each and every exception has to be handled by the programmer to avoid the program from crashing abruptly. This is done by writing additional code in a program to give proper messages or instructions to the user on encountering an exception.

This process is known as exception handling.

Need for Exception Handling:-

Exception handling is being used not only in Python but in most programming languages like C++, Java, Ruby, etc. It is a useful technique that helps in capturing runtime errors and handling them so as to avoid the program getting crashed.

Following are some of the important points regarding exception and their handlings:-

• Python categorises exceptions into distinct types so that specific exception handlers (code to handle that particular exception) can be created for each type.

• Exception handlers separate the main logic of the program from the error detection and correction code.

The segment of code where there is any possibility of error or exception, is placed inside one block.

The code to be executed in case the exception has occurred, is placed inside another block.
These statements for detection and reporting the exception do not affect the main logic of the program.

• The compiler or interpreter keeps track of the exact position where the error has occurred.

• Exception handling can be done for both user-defined and built-in exceptions.

Process of Handling Exception:-

When an error occurs, Python interpreter creates an object called the exception object.

This object contains information about the error like its type, file name and

position in the program where the error has occurred.

The object is handed over to the runtime system so that it can find an appropriate code to handle this particular exception.

This process of creating an exception object and handing it over to the runtime system is called throwing an exception.

It is important to note that when an exception occurs while executing a particular program statement, the control jumps to an exception handler, abandoning execution of the remaining program state.

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

It first searches for the method in which the error has occurred and the

exception has been raised.

If not found, then it searches the method from which this method (in which exception was raised) was called. This hierarchical search in reverse order

continues till the exception handler is found.

This entire list of methods is known as call stack. When a suitable handler is found in the call stack, it is executed by the runtime process.