

Project Development phase

Exception Handling

Date	06 May 2023
Team ID	NM2023TMID18418
Project name	CancerVision: Advanced Breast Cancer Prediction with Deep Learning

Exception Handling

The exception handling mechanism provides a means to report “exceptional circumstances” so that corrective action can be taken.

The exception handling mechanism of a separate error handling code that performs the following task: Find the problem. Inform that an error has occurred that is throw the exception.

Exceptions have the following properties:

- Use a try block around the statements that might throw exceptions.
- Once an exception occurs in the try block, the flow of control jumps to the first associated exception handler that is present anywhere in the call stack. In C#, the catch keyword is used to define an exception handler.
- If no exception handler for a given exception is present, the program stops executing with an error message.
- Don't catch an exception unless you can handle it and leave the application in a known state. If you catch `System.Exception`, rethrow it using the `throw` keyword at the end of the catch block.

-
- # Exceptions and Exception Handling

In the example, we are trying to divide a number by 0 . Here, this code generates an exception. To handle the exception, we have put the code, 5 / 0 inside the try block. Now when an exception occurs, the rest of the code inside the try block is skipped.

Exception queues are a logical stage of the payment processing where the payments are made available for further investigation or exception processing. If any exception is encountered during processing, payment transactions are moved to a queue specific to the type of exception.

It must be preceded by try block which means we can't use catch block alone. It can be followed by finally block later. The "finally" block is used to execute the necessary code of the program. It is executed whether an exception is handled or not.