EXCEPTION HANDLING

By: Ahmed Qasim

OVERVIEW

- Try, Catch, Finally
- Try Parse
- Checked, Unchecked
- Is, As

EXCEPTION HANDLING :

- It is a mechanism that allows developers to handle and respond to runtime errors in a controlled way. It is essential for building robust and error-resistant applications.
- The main goal is to catch and process errors so the program can continue running or gracefully terminate without crashing unexpectedly.

TRY PARSE

- A method to safely convert strings to numeric types without throwing exceptions.
- Used to check if the conversion is successful and retrieve the result.

TRY & CATCH

- Try Block: The try block contains the code that might throw an exception. This is where you place the code that you anticipate could potentially cause an error. If an exception occurs within the try block, the control is immediately transferred to the corresponding catch block.
- Catch Block: The catch block is used to handle exceptions that occur in the try block. You can have multiple catch blocks to handle different types of exceptions. Each catch block can catch a specific type of exception, allowing you to handle different error conditions separately.

TRY & CATCH

- Try Block:
    - Contains code that may throw an exception
- Catch Block:
    - Catches and handles exceptions
- Finally Block:
    - Executes code regardless of exceptions

CHECKED & UNCHECKED

- Checked Context:
  - Ensures safe arithmetic operations by throwing exceptions on overflow.
  - Used when accuracy and safety are critical.
- Unchecked Context:
  - Allows for faster arithmetic operations by ignoring overflow.
  - Used when performance is prioritized and overflow can be tolerated or is unlikely.

IS &AS OPERATOR

- is OperatorPurpose:
  - The is operator is used to check if an object is compatible with a given type.
  - It returns a boolean value (true or false).
- as OperatorPurpose:
  - The as operator is used to perform safe type casting.
  - It tries to convert an object to a specified type and returns null if the conversion fails, instead of throwing an exception.

EXAMPLE SCENARIO

- Scenario:
  - Divide by zero
  - ErrorHandling with try-catch

SUMMARY

- **Key Takeaways:**
  - Exception handling ensures robustness
  - Use try, catch, and finally effectively

THANKS