Multiple Exceptions & Custom Exceptions

Multiple Exceptions in Python:

- A program may face **different types of errors**.
- Each error type can be handled separately with multiple `except` blocks.

Syntax:

```
try:
   #code
except NameOfError:
   print("Error: ")
except NameOfError:
   print("Error: ")
except Exception as e:
   print("Unexpected Error:", e)
```

One Line for Multiple Exceptions:

```
try:
      x = int("abc")
except (ValueError, TypeError) as e:
   print("Error:", e)
```

- Multiple exceptions can be handled in a **single block** if the handling is the same.

Custom Exceptions:

- Sometimes we need to define **our own error** for specific conditions.
- We create a class that inherits from the built-in `Exception` class.

Example:

```
class NegativeNumberError(Exception):
   pass
def check_number(num):
   if num < 0:
      raise NegativeNumberError("Negative numbers not allowed!")
   else:
      print("Number is:", num)
try:
      check_number(-5)
except NegativeNumberError as e:
   print("Custom Exception:", e)
```

Importance:

- Prevents the program from crashing.
- Handles different error types in a clean way.
- Custom exceptions allow **specific error messages** (e.g., in banking, login, data validation).
- Improves program reliability and user experience.