Exception Handling

Exception Handling means controlling runtime errors using try-except so that the program does’t crash.

try: Code that may raise error
except: Code to run if an exception occurs
else: Code that runs when there is no exception
finally: Runs even if there is error or no error

Basic structure:

```
try:
      # Code that may raise error
except ZeroDivisionError:
       # Code to run if error occurs
```

Types of errors

1. **Syntax Errors:**
   Wrong code structure. Caught before running and code wont execute.

2. Exceptions or Runtime errors

Occurs during execution and can be handled.

Common Exceptions:

ZeroDivisionError
IndexError
KeyError
NameError
ValueError
TypeError
FileNotFoundError

- **Multiple Exception Blocks**

```
try:
   a = 5 + "k"
except TypeError:
   print("Invalid Type")
except ValueError:
   print("Invalid value")
```

- **One except handling multiple exceptions**

```
try:
    A = int(“abc”)
except (ValueError, TypeError):
   print(“Invalid”)
```

- **Raising Custom Exceptions**

```
age = 15
if age < 18:
     Raise ValueError(“Age must be 18+”)
```

- **Custom Exception class**

```
        class InvalidAgeError( Exception ):
                pass

        def check(age):
                if age < 18:
                        raise InvalidAgeError(“Age must be 18+”)

        try:
           check(12)
        except InvalidAgeError as e:
           print(e)
```

- **Exception Chaining**

When one exception leads to another

```
        try:
            int(“abc”)
        except ValueError as e:
            raise TypeError(“Invalid”) from e
```

- **Exception Inside Functions**

```
def check(a, b):
    try:
      res = a / b
    except ZeroDivisionError:
      print(“Cannot divide by 0”)
    else:
      return f”Result: {res}”
    finally:
      return “Execution Complete”
```

- Function runs code line by line. In above code if there is no exception else block runs but it does not return there itself, function prepares that return and as we know finally runs irrespective of error the return in finally overrides the else’s return so only finally is returned.

- So, returning inside finally is not recommended because it hides errors.