Introduction to Error Handling in Python

Lesson 1: What is an Error?

Explanation:

An error is a problem in a program that causes it to stop unexpectedly. There are two main types:

- Syntax Errors: Mistakes in the structure of the code.

- Runtime Errors (Exceptions): Errors that happen while the program is running.

Example:

```
if True
    print("Hello")  # Syntax Error
x = 10 / 0  # Runtime Error
```

Lesson 2: What is Exception Handling?

Explanation:

Python uses try and except blocks to catch and handle errors during execution.

Example:

```
try:
    x = 10 / 0
except ZeroDivisionError:
    print("You can't divide by zero!")
```

Lesson 3: Catching Multiple Exceptions

Explanation:

Handle different types of exceptions using multiple except blocks.

Introduction to Error Handling in Python

Example:

```
try:
    number = int(input("Enter a number: "))
    result = 10 / number
except ValueError:
    print("That's not a number!")
except ZeroDivisionError:
    print("You can't divide by zero!")
```

Lesson 4: The else and finally Blocks

Explanation:

- else: Runs if no exception occurs.
- finally: Always runs.

Example:

```
try:
    num = int(input("Enter a number: "))
except ValueError:
    print("Invalid input!")
else:
    print(f"You entered {num}")
finally:
    print("Execution complete.")
```

Introduction to Error Handling in Python

Lesson 5: Raising Your Own Exceptions

Explanation:

Use the raise keyword to create your own errors.

Example:

```
age = int(input("Enter your age: "))
if age < 0:
    raise ValueError("Age cannot be negative!")
else:
    print("Age is valid.")
```

Lesson 6: Creating Custom Exceptions (Advanced)

Explanation:

Define your own exception classes for specific error types.

Example:

```
class NegativeAgeError(Exception):
    pass

age = int(input("Enter age: "))
if age < 0:
    raise NegativeAgeError("Age must not be negative.")
```

Practice Exercises

Introduction to Error Handling in Python

1. Handle a file not found error when trying to open a file.

2. Catch both ValueError and TypeError when performing type conversion.

3. Create a calculator that handles divide-by-zero and invalid input.