

## Error Handling & Exceptions

---

### Handle Division by Zero

**Problem:** Divide two numbers entered by the user. Handle division by zero.

try:

```
a = int(input("Enter numerator: "))  
b = int(input("Enter denominator: "))  
print(a / b)
```

except ZeroDivisionError:

```
print("Error: Cannot divide by zero")
```

---

### Handle Invalid Integer Input

**Problem:** Take an integer input and handle invalid input.

try:

```
num = int(input("Enter a number: "))  
print("You entered:", num)
```

except ValueError:

```
print("Error: Invalid integer")
```

---

### Multiple Exception Handling

**Problem:** Handle both ValueError and ZeroDivisionError.

try:

```
x = int(input("Enter x: "))  
y = int(input("Enter y: "))  
print(x / y)
```

except ValueError:

```
print("Invalid input")
```

except ZeroDivisionError:

```
print("Division by zero not allowed")
```

---

### List Index Error

**Problem:** Access a list element safely.

try:

```
nums = [10, 20, 30]
```

```
print(nums[5])
```

except IndexError:

```
print("Index out of range")
```

---

### Dictionary Key Error

**Problem:** Access a dictionary key safely.

try:

```
student = {"name": "Athar"}
```

```
print(student["age"])
```

except KeyError:

```
print("Key not found")
```

---

### Type Error Handling

**Problem:** Add an integer and string safely.

try:

```
result = 10 + "5"
```

except TypeError:

```
print("Type mismatch error")
```

---

### Use finally Block

**Problem:** Show finally always executes.

try:

```
print(10 / 0)
```

except ZeroDivisionError:

```
    print("Error occurred")
finally:
    print("Execution completed")
```

---

### **File Not Found Error**

**Problem:** Read a file safely.

```
try:
    file = open("data.txt", "r")
    print(file.read())
except FileNotFoundError:
    print("File not found")
```

---

### **Input Validation**

**Problem:** Ensure input is positive.

```
try:
    n = int(input("Enter positive number: "))
    if n < 0:
        raise ValueError
    print("Valid input")
except ValueError:
    print("Invalid input")
```

---

### **Simple Calculator**

**Problem:** Divide two numbers safely.

```
try:
    a = float(input("Enter first number: "))
    b = float(input("Enter second number: "))
    print("Result:", a / b)
except ZeroDivisionError:
```

```
    print("Cannot divide by zero")
except ValueError:
    print("Invalid input")
```

---

### **Convert String to Integer**

**Problem:** Convert user input to integer safely.

```
try:
    num = int(input("Enter number: "))
except ValueError:
    print("Conversion failed")
```

---

### **Loop Until Valid Input**

**Problem:** Keep asking until valid integer is entered.

```
while True:
    try:
        num = int(input("Enter integer: "))
        break
    except ValueError:
        print("Try again")
```

---

### **Handle Multiple Errors in One Except**

**Problem:** Use a tuple in except.

```
try:
    a = int(input())
    b = int(input())
    print(a / b)
except (ValueError, ZeroDivisionError):
    print("Error occurred")
```

---

## Safe List Input

**Problem:** Add number to list safely.

```
numbers = []  
  
try:  
    numbers.append(int(input("Enter number: ")))  
except ValueError:  
    print("Invalid input")
```

---

## Raise Custom Error

**Problem:** Raise error if age < 18.

```
try:  
    age = int(input("Enter age: "))  
    if age < 18:  
        raise Exception("Not eligible")  
    print("Eligible")  
except Exception as e:  
    print(e)
```

---

## Password Validation

**Problem:** Raise error if password too short.

```
try:  
    pwd = input("Enter password: ")  
    if len(pwd) < 6:  
        raise ValueError("Password too short")  
    print("Password accepted")  
except ValueError as e:  
    print(e)
```

---

## Safe Float Conversion

**Problem:** Convert input to float.

try:

```
val = float(input("Enter value: "))
```

```
print(val)
```

except ValueError:

```
print("Invalid float")
```

---

## Math Operation with Error Handling

**Problem:** Square root safely.

```
import math
```

try:

```
num = float(input("Enter number: "))
```

```
print(math.sqrt(num))
```

except ValueError:

```
print("Cannot calculate square root of negative number")
```

---

## Division Function

**Problem:** Create a safe divide function.

```
def safe_divide(a, b):
```

```
    try:
```

```
        return a / b
```

```
    except ZeroDivisionError:
```

```
        return "Error: Division by zero"
```

```
print(safe_divide(10, 2))
```

```
print(safe_divide(10, 0))
```

---

## Login System

**Problem:** Handle invalid credentials.

try:

```
username = input("Username: ")
```

```
password = input("Password: ")
```

```
if username != "admin" or password != "1234":
```

```
    raise Exception("Invalid login")
```

```
print("Login successful")
```

except Exception as e:

```
    print(e)
```