

## **11 - Handling Exceptions**

**Ex. No. : 11.1**

**Date: 03.06.2024**

**Register No.: 231401061**

**Name: MANOJ KRISHNA R**

---

### **Invalid inputs and Out-of-range Numbers.**

Write a Python script that asks the user to enter a number within a specified range (e.g., 1 to 100). Handle exceptions for invalid inputs and out-of-range numbers.

#### **Input Format**

User inputs a number.

#### **Output Format**

Confirm the input or print an error message if it's invalid or out of range.

#### **Sample Input**

1

#### **Sample Output**

Valid input.

#### **For example:**

<b>Input</b>	<b>Result</b>
1	Valid input.
100	Valid input.
101	Error: Number out of allowed range

**Answer:**

try:

```
user_input = int(input())
```

```
if 1 <= user_input <= 100:
```

```
    print("Valid input.")
```

```
else:
```

```
    print("Error: Number out of allowed range")
```

```
except ValueError:
```

```
    print("Error: invalid literal for int()")
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
	1	Valid input.	Valid input.	
	100	Valid input.	Valid input.	
	101	Error: Number out of allowed range	Error: Number out of allowed range	

**Ex. No. : 11.2**

**Date: 03.06.2024**

**Register No.: 231401061**

**Name: MANOJ KRISHNA R**

---

### **Negative inputs and Non-Numeric Inputs**

Develop a Python program that safely calculates the square root of a number provided by the user. Handle exceptions for negative inputs and non-numeric inputs

#### **Input Format**

User inputs a number.

#### **Output Format**

Print the square root of the number or an error message if an exception occurs.

#### **Sample Input**

16

-4

#### **Sample Output**

The square root of 16.0 is 4.00

Error: Cannot calculate the square root of a negative number.

#### **For example:**

Input	Result
0	The square root of 0.0 is 0.00

**Answer:**

```
import math

def calculate_square_root():

    number = input()

    try:

        number = float(number)

        if number < 0:

            print("Error: Cannot calculate the square root of a negative number.")

        else:

            sqrt_value = math.sqrt(number)

            print(f"The square root of {number} is {sqrt_value:.2f}")

    except ValueError as e:

        print(f"Error: could not convert string to float")

calculate_square_root()
```

Input	Expected	Got
16	The square root of 16.0 is 4.00	The square root of 16.0 is 4.00
0	The square root of 0.0 is 0.00	The square root of 0.0 is 0.00
-4	Error: Cannot calculate the square root of a negative number.	Error: Cannot calculate the square root of a negative number.

**Ex. No. : 11.3**

**Date: 03.06.2024**

**Register No.: 231401061**

**Name: MANOJ KRISHNA R**

---

## **Zero and Non-Numeric Inputs**

Write a Python program that performs division and modulo operations on two numbers provided by the user. Handle division by zero and non-numeric inputs.

### **Input Format**

Two lines of input, each containing a number.

### **Output Format**

Print the result of division and modulo operation, or an error message if an exception occurs.

### **Sample Input**

7

3

### **Sample Output**

Division result: 2.3333333333333335

Modulo result: 1

### **For example:**

Input	Result
10 2	Division result: 5.0 Modulo result: 0
8 0	Error: Cannot divide or modulo by zero.

**Answer:**

```
def perform_operations():
    try:
        num1 = float(input())
        num2 = float(input())
        if num2 == 0:
            print("Error: Cannot divide or modulo by zero.")
        else:
            division_result = num1 / num2
            modulo_result = num1 % num2
            print(f'Division result: {division_result}')
            print(f'Modulo result: {int(modulo_result)}')
    except ValueError:
        print("Error: Non-numeric input provided.")
perform_operations()
```

Input	Expected	Got
10 2	Division result: 5.0 Modulo result: 0	Division result: 5.0 Modulo result: 0
7 3	Division result: 2.3333333333333335 Modulo result: 1	Division result: 2.3333333333333335 Modulo result: 1
8 0	Error: Cannot divide or modulo by zero.	Error: Cannot divide or modulo by zero.
abc 5	Error: Non-numeric input provided.	Error: Non-numeric input provided.

**Ex. No. : 11.4**

**Date: 03.06.2024**

**Register No.: 231401061**

**Name: MANOJ KRISHNA R**

---

### **Invalid Integer Inputs-1**

Write a Python program that asks the user for their age and prints a message based on the age. Ensure that the program handles cases where the input is not a valid integer.

**Input Format:** A single line input representing the user's age.

**Output Format:** Print a message based on the age or an error if the input is invalid.

**Sample Input:**

25

**Sample Output:**

You are 25 years old.

**For example:**

Input	Result
twenty	Error: Please enter a valid age.
150	You are 150 years old.
-1	Error: Please enter a valid age.



**Answer:**

try:

```
age = int(input().strip())
```

```
if age < 0:
```

```
    print("Error: Please enter a valid age.")
```

```
else:
```

```
    print(f"You are {age} years old.")
```

except ValueError:

```
    print("Error: Please enter a valid age.")
```

except EOFError:

```
    print("Error: Please enter a valid age.")
```

Input	Expected	Got
twenty	Error: Please enter a valid age.	Error: Please enter a valid age.
25	You are 25 years old.	You are 25 years old.
-1	Error: Please enter a valid age.	Error: Please enter a valid age.
150	You are 150 years old.	You are 150 years old.
	Error: Please enter a valid age.	Error: Please enter a valid age.



**Ex. No. : 11.5**

**Date: 03.06.2024**

**Register No.: 231401061**

**Name: MANOJ KRISHNA R**

---

### **Invalid Integer Inputs-2**

Write a Python program that asks the user for their age and prints a message based on the age. Ensure that the program handles cases where the input is not a valid integer.

#### **Input Format:**

A single line input representing the user's age.

#### **Output Format:**

Print a message based on the age or an error if the input is invalid.

#### **Sample Input:**

150

#### **Sample output:**

You are 150 years old.

#### **For example:**

<b>Input</b>	<b>Result</b>
rec	Error: Please enter a valid age.
25	You are 25 years old.
!@#	Error: Please enter a valid age.

**Answer:**

```
def get_age_message():  
    try:  
        age_str = input()  
        age = int(age_str)  
        if age < 0:  
            print("Error: Please enter a valid age.")  
        else:  
            print(f"You are {age} years old.")  
    except ValueError:  
        print("Error: Please enter a valid age.")  
    except EOFError:  
        print("Error: Please enter a valid age.")  
get_age_message()
```

Input	Expected	Got
25	You are 25 years old.	You are 25 years old.
rec	Error: Please enter a valid age.	Error: Please enter a valid age.
!@#	Error: Please enter a valid age.	Error: Please enter a valid age.