Lab 2 Code/Output

Sol Ben-Ishay

Code:

# Lab 2 Code

# I am calculating n factorial, but I need some changes.  
# Is there any syntax error/naming/logical error?  
# Is the code readable?  
# *FIXME*# Copy the following code and change it step by step  
# Comment out the previous one each time you complete that step.  
# Step 0  
# Original code  
# n = 5  
# for i in range(n):  
# 1x \*= i  
# print(1x)  
  
  
# Step 1  
# Fix the naming error(s) and undefined names in the above code  
# Fix naming errors and initialize them  
# Fix range values  
# n = 5  
# x = 1  
# for i in range(1, n+1):  
# x \*= i  
# print(x)  
  
  
# Step 2  
# Names like x, i are not self-descriptive  
# Rename your variables so that they are self-descriptive  
# Use Refactor --> Rename  
# factorial\_number = 5  
# result = 1  
# for value in range(1, factorial\_number+1):  
# result \*= value  
# print(result)  
  
  
# Step 3  
# The code is not flexible and configurable  
# Change the code so that n value is an input  
# Do not forget to convert input function return value to an int  
# Do not forget to provide a meaningful message for the input function  
# # Get number to find factorial of from user  
# factorial\_number = int(input("Enter the number to calculate the factorial of:"))  
#

#

#

#  
# result = 1  
# for value in range(1, factorial\_number+1):  
# result \*= value  
# print(result)

# Step 4  
# It is time to make this code modular  
# It is a good idea to make this code a function  
# Your function will be named accordingly  
# it has a parameter of type int and return an int  
# Convert the code to a function  
# Use type hinting  
# Use docstring reStructured Text style  
# def get\_factorial(number: int) -> int:  
# """Print the factorial of a number"""  
# result = 1  
# for value in range(1, number+1):  
# result \*= value  
# print(result)  
#

#  
# Call your function for a user input and print the result  
# get\_factorial(int(input("Enter the number to calculate the factorial of:")))  
  
  
# Step 5  
# Calling the function for a single input is not good enough  
# Test your function with various inputs  
# You may use a for loop to get k inputs and test your function  
# def get\_factorial(number: int) -> int:  
# """Print the factorial of a number"""  
# result = 1  
# for value in range(1, number+1):  
# result \*= value  
# print(result)  
#

#  
# # Initialize list of inputs  
# inputs = [3,4,5,7,11,15,22,26]  
#  
# # Loop through inputs with get\_factorial function  
# for input in inputs:  
# get\_factorial(input)  
  
  
# Step 6  
# Calling the function for random inputs is not good enough  
# Test your function with various inputs, like negative, positive, zero  
# You may use a for loop ranging from negatives to positives  
# def get\_factorial(number: int) -> int:  
# """Print the factorial of a number"""  
# result = 1  
# for value in range(1, number+1):  
# result \*= value  
# print(result)  
#

# # Initialize list of inputs  
# inputs = [-29,-24,-22,-7,-6,-3,0,1,2,4,7,11,15,23]

#  
# # Loop through inputs with get\_factorial function  
# for input in inputs:  
# get\_factorial(input)

# Step 7  
# Modify your function so that it will print an error message return 0 for negative numbers  
# def get\_factorial(number: int) -> int:  
# """Print the factorial of a number if it is positive, else return error"""  
# if number < 1:  
# raise ValueError("The factorial of an integer can only be calculated for integers greater than or equal to   
# 1!!")  
# else:  
# result = 1  
# for value in range(1, number+1):  
# result \*= value  
# print(result)  
#

#  
# get\_factorial(-2)  
  
  
# Step 8  
# Test your function with various inputs, like negative, positive, zero  
# You may use a list of negatives, zero and positives  
def get\_factorial(number: int) -> int:  
 *"""Print the factorial of a number if it is positive, else return error"""* if number < 1:  
 raise ValueError("The factorial of an integer can only be calculated for integers greater than or equal to 1!!")  
 else:  
 result = 1  
 for value in range(1, number+1):  
 result \*= value  
 print(result)  
  
  
# Initialize list of inputs  
inputs = [-29,-24,-22,-7,-6,-3,0,1,2,4,7,11,15,23]  
  
# Loop through inputs with get\_factorial function  
for input in inputs:  
 get\_factorial(input)

Output:

Step 1:

Text

Description automatically generated

Step 2:

Text

Description automatically generated

Step 3:

Text

Description automatically generated

Step 4:

-Get Input

Text

Description automatically generated

-Result with Input of 5Text

Description automatically generated

Step 5:Text

Description automatically generated

Step 6:Text

Description automatically generated

Step 7:Text

Description automatically generated

Step 8:

Text

Description automatically generated