

## 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):
#     lx *= i
# print(lx)

# 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:

```
/usr/local/bin/python3.9 "/Users/solbenishay/Desktop/School/2020:21/Spring/CS-120 Programming on Purpose/Lab 2/change_me_fill.py"
File "/Users/solbenishay/Desktop/School/2020:21/Spring/CS-120 Programming on Purpose/Lab 2/change_me_fill.py", line 11
    1x *= i
    ^
SyntaxError: invalid syntax

Process finished with exit code 1
```

### Step 2:

```
/usr/local/bin/python3.9 "/Users/solbenishay/Desktop/School/2020:21/Spring/CS-120 Programming on Purpose/Lab 2/change_me_fill.py"
120

Process finished with exit code 0
```

### Step 3:

```
/usr/local/bin/python3.9 "/Users/solbenishay/Desktop/School/2020:21/Spring/CS-120 Programming on Purpose/Lab 2/change_me_fill.py"
120

Process finished with exit code 0
```

### Step 4:

#### -Get Input

```
/usr/local/bin/python3.9 "/Users/solbenishay/Desktop/School/2020:21/Spring/CS-120 Programming on Purpose/Lab 2/change_me_fill.py"
Enter the number to calculate the factorial of:
```

## -Result with Input of 5

```
/usr/local/bin/python3.9 "/Users/solbenishay/Desktop/School/2020:21/Spring/CS-120 Programming on Purpose/Lab 2/change_me_fill.py"
Enter the number to calculate the factorial of:
120

Process finished with exit code 0
|
```

## Step 5:

```
/usr/local/bin/python3.9 "/Users/solbenishay/Desktop/School/2020:21/Spring/CS-120 Programming on Purpose/Lab 2/change_me_fill.py"
6
24
120
5040
39916800
1307674368000
112400072777607680000
403291461126605635584000000

Process finished with exit code 0
```

## Step 6:

```
/usr/local/bin/python3.9 "/Users/solbenishay/Desktop/School/2020:21/Spring/CS-120 Programming on Purpose/Lab 2/change_me_fill.py"
1
1
1
1
1
1
1
1
1
2
24
5040
39916800
1307674368000
25852016738884976640000

Process finished with exit code 0
|
```

## Step 7:

```
/usr/local/bin/python3.9 "/Users/solbenishay/Desktop/School/2020:21/Spring/CS-120 Programming on Purpose/Lab 2/change_me_fill.py"
Traceback (most recent call last):
  File "/Users/solbenishay/Desktop/School/2020:21/Spring/CS-120 Programming on Purpose/Lab 2/change_me_fill.py", line 122, in <module>
    get_factorial(-2)
  File "/Users/solbenishay/Desktop/School/2020:21/Spring/CS-120 Programming on Purpose/Lab 2/change_me_fill.py", line 114, in get_factorial
    raise ValueError("The factorial of an integer can only be calculated for integers greater than or equal to 1!!")
ValueError: The factorial of an integer can only be calculated for integers greater than or equal to 1!!

Process finished with exit code 1
```

## Step 8:

```
/usr/local/bin/python3.9 "/Users/solbenishay/Desktop/School/2020:21/Spring/CS-120 Programming on Purpose/Lab 2/change_me_fill.py"
Traceback (most recent call last):
  File "/Users/solbenishay/Desktop/School/2020:21/Spring/CS-120 Programming on Purpose/Lab 2/change_me_fill.py", line 144, in <module>
    get_factorial(input)
  File "/Users/solbenishay/Desktop/School/2020:21/Spring/CS-120 Programming on Purpose/Lab 2/change_me_fill.py", line 131, in get_factorial
    raise ValueError("The factorial of an integer can only be calculated for integers greater than or equal to 1!!")
ValueError: The factorial of an integer can only be calculated for integers greater than or equal to 1!!

Process finished with exit code 1
```