

Daryl D. Silva

BSCpE-2B2

Laboratory Exercise No. 2 CH2

Title: Understanding Programming Constructs

Brief Introduction

Programming constructs are the building blocks of any software application. These include variables, loops, conditionals, functions, data structures, and recursion. This lab focuses on these constructs and their implementation in Python.

Procedure

1. Create a new Python file in VS Code.
2. Implement examples of variables, loops, conditionals, and recursion.

Example Code:

```
# Variables and loops
```

```
nums = [1, 2, 3, 4, 5]
```

```
for num in nums:
```

```
    print(num)
```

```
# Conditional Statements
```

```
if len(nums) > 3:
```

```
    print("List is large")
```

```
else:
```

```
    print("List is small")
```

```
# Recursion
```

```
def factorial(n):
```

```
    if n == 0:
```

```
        return 1
```

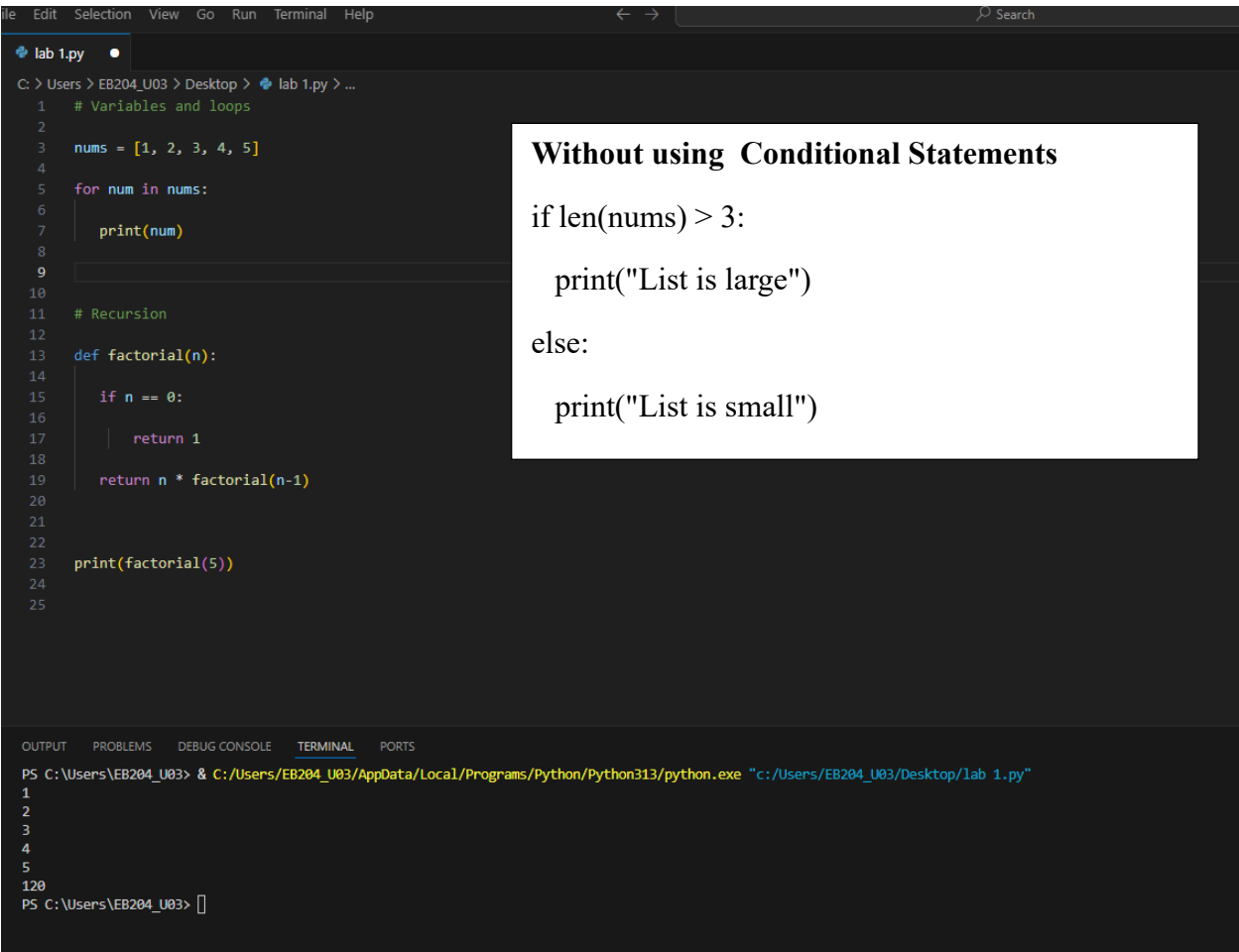
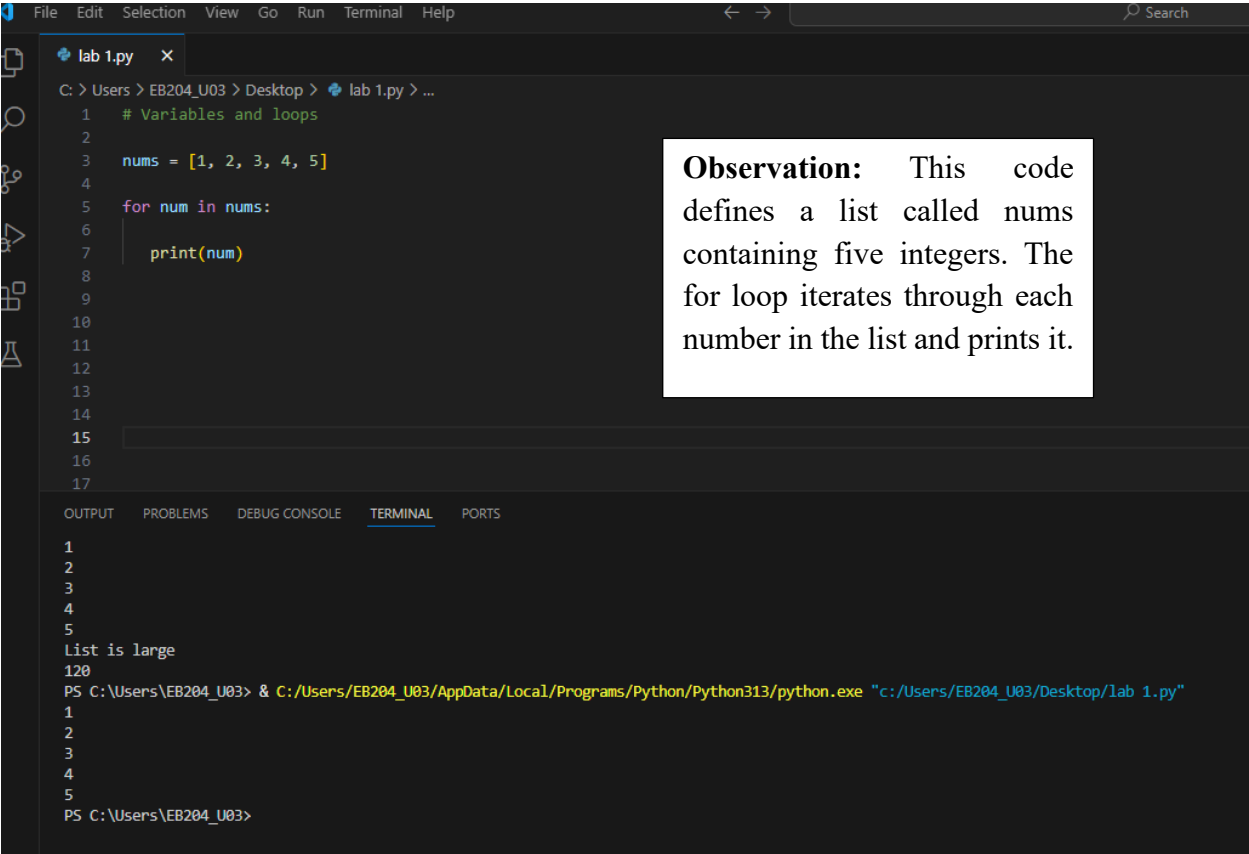
```
    return n * factorial(n-1)
```

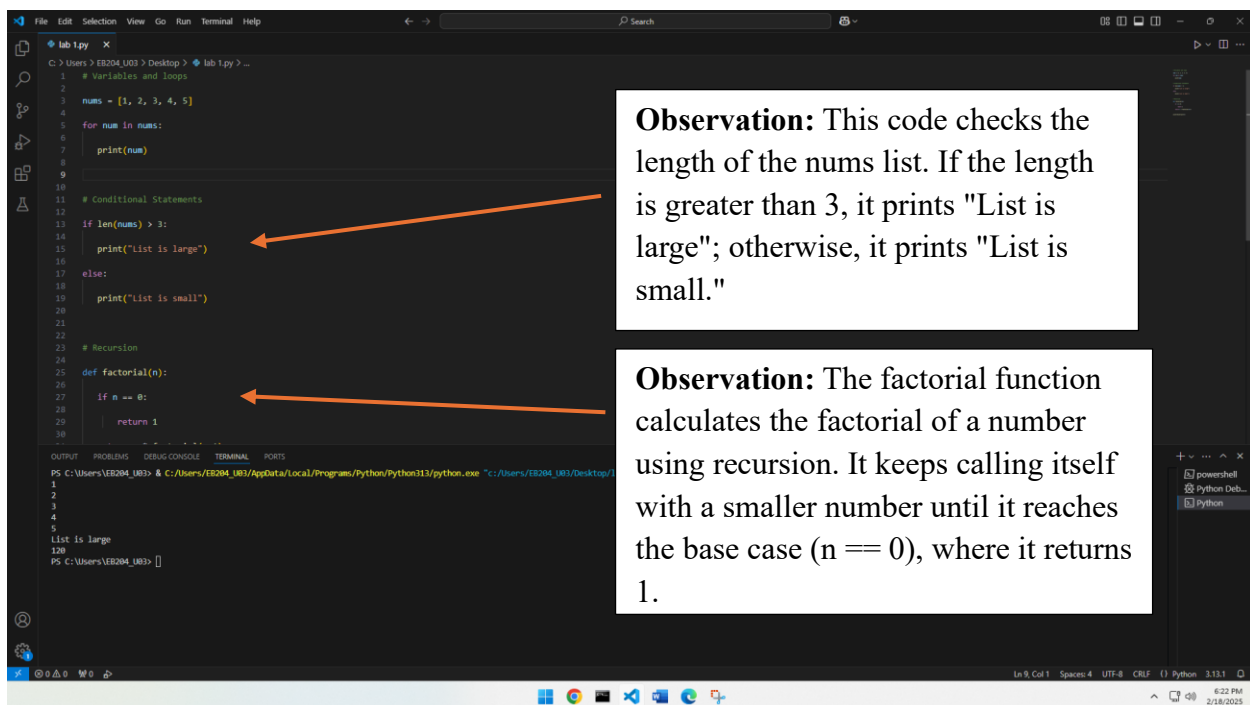
```
print(factorial(5))
```

1. Run the program and observe the results.

Results

Record outputs for different test cases.





Follow-Up Questions

1. What is the purpose of using recursion?

Answer: Recursion is used to solve problems that can be broken down into smaller, similar subproblems. It simplifies the code for tasks like tree traversal, factorial calculation, or searching algorithms, where the same function repeatedly calls itself with smaller inputs until a base case is reached.

2. How do loops differ from recursion?

- Answer:
- Loops repeat a block of code a specific number of times or while a condition holds true, typically using constructs like for or while.
 - Recursion calls a function within itself, breaking a problem into smaller subproblems. Each recursive call processes part of the problem and eventually reaches a base case to stop the recursion.
 - Loops are usually more efficient in terms of memory, while recursion can be more intuitive and elegant for problems with a natural recursive structure (e.g., tree traversal).

3. Explain a scenario where conditionals are essential.

Answer: In an **e-commerce application**, conditionals are essential when checking if a user is logged in. For instance, an if statement might be used to check if a user is authenticated before allowing them to add items to their shopping cart or proceed to checkout. Without conditionals, actions could be executed regardless of user status, leading to errors or undesired behavior.

Findings

Demonstrate understanding of the constructs through written observations.

Answer: Each construct serves a unique purpose: loops for repetition, conditionals for decision-making, and recursion for solving problems that can be broken into smaller subproblems.

Summary

Programming constructs are fundamental building blocks that help developers solve problems and create efficient software. The key constructs explored in this example are **variables**, **loops**, **conditional statements**, and **recursion**. Each plays a specific role in how programs execute:

- **Variables** store data and allow us to manipulate values in the program.
- **Loops** repeat actions a set number of times or while a condition is true, making it easier to process data iteratively.
- **Conditional Statements** help the program make decisions by evaluating conditions and executing different blocks of code based on the outcome.
- **Recursion** allows functions to call themselves to solve problems by breaking them into smaller, manageable subproblems, particularly useful in tasks like calculating factorials or traversing complex data structures.

Conclusion

Understanding programming constructs is essential for creating efficient, readable, and maintainable code. These constructs help developers manage data processing, decision-making, repetition, and problem-solving. Mastery of these foundational concepts enables the creation of high-quality software and serves as the foundation for more advanced programming tasks. Constructs like loops, conditionals, and recursion are crucial for building intuitive and powerful software.