

Lab 1: Basic For Loop in Python

Objective:

Understand how to use a `for` loop to iterate through a sequence of numbers.

Scenario:

You are tasked to create a program that prints all even numbers from 1 to 20.

Instructions:

1. Use a `for` loop to iterate through numbers from 1 to 20.
 2. Inside the loop, use an `if` statement to check if the number is even.
 3. Print the even numbers.
-

Code Template:

```
# Basic for loop to find even numbers
for num in range(1, 21): # Loop through numbers 1 to 20
    if num % 2 == 0: # Check if the number is even
        print(num)
```

Explanation:

1. The `range(1, 21)` function generates numbers from 1 to 20.
 2. The `for` loop iterates through each number in this range.
 3. The `if` statement checks if a number is divisible by 2 (`num % 2 == 0`).
 4. If the condition is `True`, the number is printed.
-

Sample Output:

Copy code

```
2
4
6
8
10
12
```

14
16
18
20

Lab 2: Advanced While Loop in Python

Objective:

Learn to use a `while` loop to create a program with user-controlled termination.

Scenario:

You are creating a simple calculator that repeatedly asks the user for two numbers and an operation (+, -, *, /). The program should stop when the user types "exit".

Instructions:

1. Use a `while` loop to continuously prompt the user for inputs.
 2. Allow the user to enter "exit" to terminate the program.
 3. Perform the operation based on the user's input and display the result.
 4. Handle invalid inputs gracefully.
-

Code Template:

```
# Advanced while loop for a calculator
while True: # Infinite loop
    # Ask the user for input
    operation = input("Enter operation (+, -, *, /) or 'exit' to quit: ")
    operation = operation.strip()

    # Check for exit condition
    if operation.lower() == "exit":
        print("Calculator exiting. Goodbye!")
        break

    # Get two numbers from the user
    try:
        num1 = float(input("Enter the first number: "))
        num2 = float(input("Enter the second number: "))
    except ValueError:
        print("Invalid input! Please enter numeric values.")
```

```
        continue

    # Perform the operation
    if operation == "+":
        print(f"The result is: {num1 + num2}")
    elif operation == "-":
        print(f"The result is: {num1 - num2}")
    elif operation == "*":
        print(f"The result is: {num1 * num2}")
    elif operation == "/":
        if num2 != 0:
            print(f"The result is: {num1 / num2}")
        else:
            print("Error: Division by zero is not allowed.")
    else:
        print("Invalid operation! Please enter one of (+, -, *, /).")
```

Explanation:

1. A `while True` loop creates an infinite loop that will continue until broken.
 2. The `input()` function collects user input for the operation and two numbers.
 3. The `if` condition checks for the "exit" command and terminates the loop using `break`.
 4. `try-except` handles invalid numeric inputs gracefully.
 5. Nested `if-elif` statements perform the specified operation.
 6. Division by zero is explicitly handled to avoid runtime errors.
-

Sample Runs:

Run 1:

```
Enter operation (+, -, *, /) or 'exit' to quit: +
Enter the first number: 10
Enter the second number: 5
The result is: 15.0
```

Run 2:

```
Enter operation (+, -, *, /) or 'exit' to quit: /
Enter the first number: 10
Enter the second number: 0
Error: Division by zero is not allowed.
```

Run 3:

```
Enter operation (+, -, *, /) or 'exit' to quit: exit
Calculator exiting. Goodbye!
```