Lab: Repetition Structures (Loops)

Lab Outcomes

By the end of this lab, you will be able to:

- Write correct `while` and `for` loops with proper indentation.
- Explain what an **iteration** is and trace loop execution step-by-step.
- Use `range(start, stop, step)` without off-by-one mistakes.
- Build **accumulators** (running totals) and counters correctly.
- Use **sentinel** loops and **input validation** loops.
- Debug and correct common loop errors (infinite loops, wrong update, wrong range, wrong accumulator).

Suggested 4-Hour Lab Plan

- Warm-up examples + `while` basics + tracing iterations
- `for` + `range()` + patterns
- Accumulators, sentinel loops, validation loops
- Debugging drills + mini-project exercises (30 total)

How to Debug Loop Code (Fast Checklist)

If your loop gives wrong output or runs forever:

1. Check indentation (is the body inside the loop?).
2. Check initialization (counter/total set correctly before loop?).
3. Check condition (will it eventually become False?).
4. Check update (counter changes every iteration?).
5. For `range()`, check **stop is excluded** and step sign.

How Iterations Work (Mental Model)

Every loop repeats this cycle:

1. **Initialize** (starting values)
2. **Test** / **Next value** (while tests condition; for gets next value)
3. **Body** (work done once per iteration)
4. **Update** (move toward stopping)

Iteration = one complete execution of the loop body.

1: while Counter (1 to 5)

```
count = 1                              # Initialize counter at 1

while count <= 5:                      # Condition checked before each
    iteration
    print("count =", count)            # Body: prints current count
    count = count + 1                  # Update: moves count toward
        stopping
```

1 Trace (What Happens Each Iteration?)

For the previous code:

- Iteration 1: count=1 prints 1, then count becomes 2
- Iteration 2: count=2 prints 2, then count becomes 3
- Iteration 3: count=3 prints 3, then count becomes 4
- Iteration 4: count=4 prints 4, then count becomes 5
- Iteration 5: count=5 prints 5, then count becomes 6
- Next test: $6 \leq 5$ is False $\rightarrow$ loop stops

2: Common Infinite Loop + Fix

Buggy: missing update (infinite loop)

```
count = 1                                  # Initialize counter
while count <= 5:                          # Condition stays True forever (
    count never changes)
    print(count)                           # Prints forever
```

Fix: add update

```
count = 1                                  # Initialize counter
while count <= 5:                          # Condition can become False
    print(count)                           # Print current count
    count += 1                             # Update counter each iteration
```

3: for + range (1 to 5)

```
for i in range(1, 6):                   # range(1,6) generates 1,2,3,4,5
    print("i =", i)                     # i gets a new value each
       iteration
```

4: range Variants

```
for x in range(5):                    # 0,1,2,3,4
    print(x)                          # Prints 0 to 4

for x in range(2, 7):                 # 2,3,4,5,6
    print(x)                          # Prints 2 to 6

for x in range(10, 0, -2):            # 10,8,6,4,2
    print(x)                          # Descending needs negative step
```

5: Accumulator (Sum of 5 Numbers)

```
total = 0.0                                    # Accumulator must start at 0
for _ in range(5):                             # Repeat exactly 5 times
    n = float(input("Enter number: "))   # Read one number
    total += n                                 # Add to running total each
        iteration
print("Total =", total)                        # Print final accumulated total
```

6: Sentinel Loop (Stop with 0)

```
total = 0                                    # Accumulator for sum
x = int(input("Enter a number (0 to stop): "))   # Priming read

while x != 0:                                # Sentinel condition (0 ends
    input)
    total += x                               # Add current number to sum
    x = int(input("Enter a number (0 to stop): "))   # Read next
        number

print("Sum =", total)                        # Final sum after loop ends
```

7: Input Validation Loop (0..100)

```
score = int(input("Enter score (0-100): "))   # Priming read

while score < 0 or score > 100:               # Invalid range
    condition
    print("ERROR: score must be 0..100")      # Message for user
    score = int(input("Enter score (0-100): "))  # Read again

print("Accepted score:", score)               # Safe to use now
```

8: Nested Loops (Rectangle Pattern)

```
rows = 4                                          # Number of rows
cols = 6                                          # Number of columns

for r in range(rows):                             # Outer loop controls
    rows
    for c in range(cols):                         # Inner loop controls
        columns
        print("*", end="")                        # Print without newline
    print()                                       # Newline after each
        row
```

9: Optional Turtle Loop (Square)

```
import turtle                                    # Turtle graphics
    module

for _ in range(4):                               # 4 iterations for 4
    sides
    turtle.forward(100)                          # Move forward 100
        pixels
    turtle.right(90)                             # Turn right 90 degrees
```

Mini-Quiz (Predict Before Running)

Predict outputs:

1. `for i in range(3):  print(i)`
2. `for i in range(1,4):  print(i)`
3. `for i in range(5,0,-1):  print(i)`
4. `total=0; for i in range(1,6):  total+=i; print(total)`

Exercise 01: Print 1 to 10 using while

Write a program that prints numbers 1 to 10 using a `while` loop.

Solution 01

```
i = 1                                  # Initialize counter at 1
while i <= 10:                         # Repeat while i is at most 10
    print(i)                           # Print current i (one iteration
        output)
    i += 1                             # Update i so the loop can stop
```

Exercise 02: Print even numbers 2 to 20 (while)

Print: 2,4,6,...,20 using a `while` loop.

Solution 02

```
n = 2                                      # Start from first even number
while n <= 20:                             # Continue until 20
    print(n)                               # Print current even number
    n += 2                                 # Jump to next even number
```

Exercise 03: Sum 1..n (for)

Input `n`. Compute and print $\sum_{k=1}^{n} k$ using `for` + `range`.

Solution 03

```
n = int(input("Enter n: "))                  # Read n as integer
total = 0                                    # Accumulator for sum (start at
    0)

for k in range(1, n + 1):                    # k = 1,2,...,n
    total += k                               # Add k into running total

print("Sum =", total)                        # Print final sum
```

Exercise 04: Factorial n (for)

Input `n`. Compute $n!$ using a loop (do not use math.factorial).

Solution 04

```
n = int(input("Enter n: "))              # Read n
fact = 1                                 # Multiplicative accumulator
    starts at 1

for k in range(1, n + 1):                # Multiply by 1..n
    fact *= k                            # Update factorial

print("n! =", fact)                      # Print factorial
```

Exercise 05: Count digits (while)

Input a positive integer and count how many digits it has using `while`.

Solution 05

```
num = int(input("Enter a positive integer: "))  # Read integer
count = 0                                       # Digit counter
    starts at 0

while num > 0:                         # Repeat until number becomes 0
    num //= 10                         # Remove last digit (integer
        division)
    count += 1                         # Count one digit removed

print("Digits =", count)               # Print digit count
```

Exercise 06: Reverse a number (while)

Input a positive integer and print its reverse using // and %.

Solution 06

```
num = int(input("Enter a positive integer: "))  # Read integer
rev = 0                                         # Reversed number
    starts at 0

while num > 0:                      # Repeat until all digits
    processed
    digit = num % 10                # Extract last digit
    rev = rev * 10 + digit          # Append digit to reverse
    num //= 10                      # Remove last digit

print("Reverse =", rev)             # Print reversed number
```

Exercise 07: Sentinel sum (0 ends)

Keep taking integers from user and add to sum until user enters 0. Print sum.

Solution 07

```
total = 0                                        # Accumulator
    for sum
x = int(input("Enter integer (0 to stop): "))   # Priming read

while x != 0:                                    # Sentinel loop
    continues until 0
    total += x                                   # Add current
        value
    x = int(input("Enter integer (0 to stop): "))  # Read next
        value

print("Sum =", total)                            # Print final
    sum
```

Exercise 08: Sentinel average (-1 ends)

Input numbers until user types -1. Print average (handle case: no numbers).

Solution 08

```
total = 0.0                                         # Sum accumulator
count = 0                                           # Counter for how
    many numbers
x = float(input("Enter number (-1 to stop): "))     # Priming read

while x != -1:                                      # Sentinel is -1
    total += x                                      # Add into total
    count += 1                                      # Count this
        number
    x = float(input("Enter number (-1 to stop): "))# Read next

if count == 0:                                      # Avoid division
   by zero
    print("No numbers entered.")                    # Special case
else:
    avg = total / count                             # Compute average
    print("Average =", avg)                         # Print average
```

Exercise 09: Validate score (0..100)

Input score. If invalid, keep asking until valid. Then print accepted score.

Solution 09

```
score = int(input("Enter score (0-100): "))         # Priming read

while score < 0 or score > 100:                     # Invalid
   condition
    print("ERROR: Score must be 0..100")            # Message
    score = int(input("Enter score (0-100): "))     # Read again

print("Accepted score:", score)                     # Valid score
```

Exercise 10: Validate positive integer

Input an integer. Keep asking until user enters a positive integer (>0).

Solution 10

```
n = int(input("Enter a positive integer: "))       # Priming read

while n <= 0:                                       # Invalid if 0 or
    negative
    print("ERROR: must be > 0")                     # Tell the user
    n = int(input("Enter a positive integer: "))    # Read again

print("Accepted:", n)                               # Valid value
```

Exercise 11: Multiplication table of 5

Using a loop, print: 5x1=5 ... 5x10=50.

Solution 11

```
n = 5                                    # Fixed number for table

for i in range(1, 11):                   # i = 1..10
    product = n * i                      # Compute 5*i
    print(n, "x", i, "=", product)       # Print one table line
```

Exercise 12: Rectangle pattern (nested loops)

Print a 4x6 rectangle of $*$ using nested loops.

Solution 12

```
rows = 4                                  # Total rows
cols = 6                                  # Total columns

for r in range(rows):                     # Outer loop: each row
    for c in range(cols):                 # Inner loop: each column
        print("*", end="")                # Print star without
            newline
    print()                               # Newline after each row
```

Exercise 13: Right triangle of stars

Print this using loops (size 5):

```
*
**
***
****
*****
```

Solution 13

```
size = 5                                             # Number of rows in
    triangle

for r in range(1, size + 1):                         # r = 1..5 (stars in row)
    for _ in range(r):                               # Print r stars
        print("*", end="")                           # Same line
    print()                                          # Newline after each row
```

Exercise 14: Stair pattern with spaces +

Print:

```
#
 #
  #
   #
    #
```

(using 5 rows)

Solution 14

```
rows = 5                                         # Total rows

for r in range(rows):                            # Row index 0..4
    for _ in range(r):                           # Print r spaces
        print(" ", end="")                       # Spaces shift the #
    print("#")                                   # Print # and newline
```

Exercise 15: Print coordinates (nested loops)

Print pairs (r,c) for r=1..3 and c=1..4 using nested loops.

Solution 15

```
for r in range(1, 4):                                # r = 1..3
    for c in range(1, 5):                            # c = 1..4
        print("(", r, ",", c, ")", sep="") # Print coordinate pair
```

Exercise 16: Countdown with range step

Print 10 down to 1 using `for` and `range` with negative step.

Solution 16

```
for n in range(10, 0, -1):                    # 10,9,8,...,1 (stop=0
    excluded)
    print(n)                                  # Print each countdown
        number
```

Exercise 17: Multiples of 7 up to 70

Print: 7, 14, 21, ..., 70 using `range`.

Solution 17

```
for n in range(7, 71, 7):                    # Start=7, stop=71 (to
    include 70), step=7
    print(n)                                 # Print each multiple of
        7
```

Exercise 18: Compute x^n without **

Input x and n. Compute x^n using a loop (multiplication repeated).

Solution 18

```
x = float(input("Enter x: "))                  # Base value
n = int(input("Enter n (>=0): "))              # Exponent as integer

power = 1                                      # Start with
    multiplicative identity
for _ in range(n):                             # Repeat n times
    power *= x                                 # Multiply by x each
        iteration

print("x^n =", power)                          # Print computed power
```

Exercise 19: Average of 5 numbers (for)

Read 5 numbers using a loop and print their average.

Solution 19

```
total = 0.0                                          # Sum accumulator
count = 5                                             # Fixed count

for _ in range(count):                                # Repeat 5 times
    n = float(input("Enter number: "))                # Read a number
    total += n                                        # Add to total

avg = total / count                                   # Average = total / 5
print("Average =", avg)                               # Print average
```

Exercise 20 (Debug): Fix the infinite loop

This code runs forever. Fix it:

```
count = 1
while count <= 5:
    print(count)
```

Solution 20 (Corrected Code)

```
count = 1                                    # Initialize counter
while count <= 5:                            # Stop condition when count
    becomes 6
     print(count)                            # Print current count
     count += 1                              # FIX: update count each
         iteration
```

Exercise 21 (Debug): Off-by-one in range

This prints 1..4 but we want 1..5. Fix it:

```
for i in range(1, 5):
    print(i)
```

Solution 21 (Corrected Code)

```
for i in range(1, 6):                          # FIX: stop must be 6 to include
    5
    print(i)                                   # Prints 1,2,3,4,5
```

Exercise 22 (Debug): Wrong accumulator start

This tries to sum 1..5, but total starts wrong. Fix it:

```
total = 100
for i in range(1, 6):
    total += i
print(total)
```

Solution 22 (Corrected Code)

```
total = 0                                    # FIX: accumulator must start
    at 0
for i in range(1, 6):                        # i = 1..5
    total += i                               # Add each i to total
print("Sum 1..5 =", total)                   # Prints 15
```

Exercise 23 (Debug): Sentinel loop missing priming read

Fix this sentinel loop (it uses x before reading it):

```
total = 0
while x != 0:
    total += x
    x = int(input("Enter (0 to stop): "))
print(total)
```

Solution 23 (Corrected Code)

```
total = 0                                         # Sum accumulator
x = int(input("Enter (0 to stop): "))             # FIX: priming read
    BEFORE loop

while x != 0:                                     # Sentinel condition
    total += x                                    # Add current input
    x = int(input("Enter (0 to stop): "))         # Read next input

print("Sum =", total)                             # Print final sum
```

Exercise 24 (Debug): Pattern prints in one line

Fix so it prints a 3x4 rectangle:

```
for r in range(3):
    for c in range(4):
        print("*", end="")
```

Solution 24 (Corrected Code)

```
for r in range(3):                        # 3 rows
    for c in range(4):                    # 4 columns
        print("*", end="")                # Print stars on same line
    print()                               # FIX: newline after each row
```

Exercise 25: Minimum until sentinel 0

Read integers until 0. Print the minimum (assume at least one number entered).

Solution 25

```
x = int(input("Enter integer (0 to stop): "))    # Read first value (
    assume not 0)
min_val = x                                      # Initialize minimum
     as first value

x = int(input("Enter integer (0 to stop): "))    # Read next value
while x != 0:                                    # Stop when 0
    entered
    if x < min_val:                              # If current is
        smaller
        min_val = x                              # Update minimum
    x = int(input("Enter integer (0 to stop): "))# Read next

print("Minimum =", min_val)                      # Print minimum
```

Exercise 26: Maximum until sentinel 0

Read integers until 0. Print the maximum (assume at least one number entered).

Solution 26

```
x = int(input("Enter integer (0 to stop): "))    # Read first value (
    assume not 0)
max_val = x                                       # Initialize maximum

x = int(input("Enter integer (0 to stop): "))    # Read next value
while x != 0:                                     # Loop until
    sentinel
    if x > max_val:                               # If current is
        larger
        max_val = x                               # Update maximum
    x = int(input("Enter integer (0 to stop): "))# Read next

print("Maximum =", max_val)                       # Print maximum
```

Exercise 27: Sum of squares 1..n

Input n. Compute $1^2 + 2^2 + \cdots + n^2$.

Solution 27

```
n = int(input("Enter n: "))             # Read n
total = 0                               # Accumulator for sum of squares

for k in range(1, n + 1):               # k = 1..n
    total += k * k                      # Add k^2 to total

print("Sum of squares =", total)    # Print result
```

Exercise 28: Count vowels in a string (for)

Input a string and count vowels (a,e,i,o,u) using a `for` loop.

Solution 28

```
s = input("Enter a string: ")          # Read input string
count = 0                              # Vowel counter

for ch in s:                           # Iterate over each character (
    one iteration per char)
    if ch in "aeiouAEIOU":             # Check if char is a vowel
        count += 1                     # Increase count

print("Vowels =", count)               # Print vowel count
```

Exercise 29: Running total of distances

Given list: `[1.2, 0.5, 3.0, 2.1]` km. Use a loop to compute total distance.

Solution 29

```
distances = [1.2, 0.5, 3.0, 2.1]    # List of distances in km
total = 0.0                          # Accumulator for total distance

for d in distances:                  # Each iteration takes one list
    item
    total += d                       # Add to running total

print("Total distance (km) =", total)  # Print total
```

Exercise 30: Mini-project (Guessing loop)

Secret number is 7. Keep asking until user guesses correctly. Count attempts.

Solution 30

```
secret = 7                                       # Fixed secret number
attempts = 0                                     # Counter for attempts

guess = int(input("Guess the number: "))         # Priming read
attempts += 1                                    # First attempt counted

while guess != secret:                           # Loop until guess is
    correct
    print("Wrong! Try again.")                   # Feedback message
    guess = int(input("Guess the number: "))     # Read next guess
    attempts += 1                                # Count each attempt

print("Correct!")                                # Success message
print("Attempts =", attempts)                    # Print total attempts
```