Department of Computer Science

FACULTY OF SCIENCE
SESSION (2025-2026)

ALIGARH MUSLIM UNIVERSITY

Laboratory Course-VI
Course Code- CABSMJ6P06

Submitted By -
Name – Syed Hasan Ifham Zaidi
Enrollment no- GL4988
Faculty no - *22 CABSA141*

Submitted to- >

INDEX

WEEK NO	PROBLEMS WITH DESCRIPTION		PAGE NO.	SIGNATURE OF THE TEACHER WITH DATE
1	1#	Discuss the process by which Python executes a program, focusing on the stages involved from source.		
	2#	Discuss the fundamental features of the Python programming language that make it widely used for modern software		
	3#	Install IDLE interpreter for Python programming by following the mentioned		
2	1#	Use interactive Shell to print: User said "Great app!" and directory paths.		
	2#	A shopkeeper uses Python shell for quick billing calculations. Evaluate various arithmetic expressions (addition, subtraction, multiplication, division, modulus).		
	3#	A college form stores student names. Create variables firstName and lastName with your own names.		
	4#	Print firstName and lastName in a single line using f-strings.		
	5#	A heritage museum kiosk needs an automated display-and-billing script. Create a single Python file kioskSystem.py that performs billing calculations and displays information		
	6#	An engineer calculates load using a polynomial. Given specific variable values, compute the result of the algebraic expression.		
	7#	Write a program to compute the value of a specific algebraic expression where x and y are read using the input() function.		

WEEK NO	PROBLEMS WITH DESCRIPTION		PAGE NO.	SIGNATURE OF THE TEACHER WITH DATE
	8#	A technician optimizes multiplication using bitwise operations. Read an integer and multiply it by 2 using the left shift operator		
3	1#	A bookstore's billing app needs a quick utility script. Write a Python program that reads N integers (prices) and computes: Sum, Product,		
	2#	A weather station needs temperature conversion. Write a program to convert		
	3#	An architecture tool calculates dimensions. Write a program to compute the Surface Area and Volume of a Cuboid given length, breadth,		
	4#	Write a program to add two numbers without using arithmetic operators, using bitwise		
	5#	A hardware technician tests divider circuits. Read an integer and divide it by 4 using the bitwise >> operator.		
	6#	A robotics controller swaps sensor IDs efficiently. Read two integers and exchange their values using the XOR (^) operator.		
	7#	A gaming console optimizes score calculation. Multiply a number by 10 using bitwise		
	8#	A data-logging device swaps configuration settings. Read two integers and exchange their values using addition and subtraction (avoid using a third variable).		
	9#	A biology simulation needs repeated multiplication. Write a program to compute the factorial of an integer using a loop.		
	10#	A math module checks sequence validity. Write a program to determine whether a number is a Fibonacci number.		

WEEK NO		PROBLEMS WITH DESCRIPTION	PAGE NO.	SIGNATURE OF THE TEACHER WITH DATE
	11#	A security system verifies prime-number-based keys. Write a program to check whether an integer is prime, or else output its first		
	12#	A signal-processing unit finds harmony between two frequencies. Write a program to compute the GCD of two numbers.		
4	1#	A teacher wants to automate small mathematical tasks. Define functions to: compute average of marks, convert temperature, and calculate perimeter of a		
	2#	A science lab technician is calculating the volume of spherical containers. Write a function that accepts the radius and returns the volume of the sphere.		
	3#	A landscaping company maintains circular flower beds. Write a function that takes outer radius R and inner radius r, validates them, and returns the effective usable area.		
	4#	A data analyst needs factorial values to compute permutations. Write a Python function to calculate the factorial of a non-		
	5#	An accountant wants a utility that extracts a digit-sum check value. Write a Python function that returns the sum of digits of a		
	6#	A math research assistant is verifying test values. Write a Python function to check whether a number is a Perfect Number.		
	7#	Write a Python function to check whether a given number is a prime number or not.		
	8#	A text-analysis tool needs statistics from a sentence. Write a function that accepts a string and counts the number of uppercase and		

WEEK NO		PROBLEMS WITH DESCRIPTION	PAGE NO.	SIGNATURE OF THE TEACHER WITH DATE
5	1#	A data automation team is creating a utility module. Write a program that accepts inputs and uses functions to: find largest of three numbers, compute volume of shapes, area of rectangle, circumference of circle, swap variables, and find distance between points.		
	2#	A billing system receives an unknown count of numeric entries. Write a function that accepts arbitrary integers (*args) and returns		
	3#	A text-filtering tool needs to remove noisy characters. Write a program that takes a string and removes all characters at even index		
	4#	Write a Python script that takes input from the user and displays that input back in upper and lower cases.		
	5#	A simple analytics module must calculate keyword frequency. Write a program to count occurrences of each word in a sentence.		
	6#	A document cleanup tool needs to normalize text. Write a program to remove indentation from each line of a multi-line text.		
	7#	Write a function that reverses a string by modifying the input array in-place with O(1)		
	8#	Given a string, reverse the order of characters in each word within a sentence while preserving whitespace and word order.		
	9#	A parser converts messages into tokens. Write a program to convert a string into a list of		
	10#	Write a Python program to count and display all vowels present in the text, as well as consonants and their count.		

WEEK NO	PROBLEMS WITH DESCRIPTION		PAGE NO.	SIGNATURE OF THE TEACHER WITH DATE
6	1#	A grading system compares three internal-assessment scores. Write a program to find the highest score among the three using only if statements.		
	2#	A sensor module collects readings from three probes. Write a program to determine the lowest reading using if–else logic.		
	3#	Write a program to check if a year is a leap		
	4#	Write a program to print the number of days in a month using chained conditionals.		
	5#	Write a program that takes a month and year, and prints the correct number of days, accounting for leap years.		
	6#	An architectural software checks triangular panels. Write a program to compute area and identify if a triangle is equilateral, isosceles, or		
	7#	A device logs data indexes. Given two numbers r1 and r2, create a list of integers from r1 to r2 using range().		
	8#	A parser receives a mixed list. Write a program to extract and add only numeric items using isinstance().		
	9#	A game engine verifies level IDs. Write a program to check whether a given list of numbers is consecutive.		
	10#	FizzBuzz: Given integer n, return a string array where multiples of 3 are "Fizz", 5 are "Buzz", and both are "FizzBuzz"		
	1#	A cryptographic security module uses "Narcissistic Numbers". Write a program using a while loop to check if a numeric key is		

WEEK NO		PROBLEMS WITH DESCRIPTION	PAGE NO.	SIGNATURE OF THE TEACHER WITH DATE
7	2#	A biological simulation tool models cell growth. Write a recursive function to generate the first n terms of the Fibonacci sequence.		
	3#	A data science team needs to calculate permutations. Write a recursive function calculate_routes(n) to compute the factorial of a number.		
	4#	A telecommunications engineer synchronizes signals. Write a recursive function find_signal_sync(a, b) to find the GCD using the Euclidean algorithm.		
	5#	Write a program to print a specific inverted right-angle triangle pattern using a for loop.		
	6#	A network packet analyzer receives packet IDs. Filter these into two lists: even IDs (low priority) and odd IDs (high priority).		
	7#	Given an integer x, return true if x is a palindrome, and false otherwise.		
	8#	An inventory management system consolidates records. Merge two lists of product IDs and display the final list in sorted		
	9#	An algorithm visualization tool demonstrates sorting. Write a program to find the second largest number in a list using the Bubble Sort		
	1#	A file management system needs to organize filenames. Write a program to sort a list of strings based on string length.		
	2#	A marketing team has two email lists. Create a master list containing all unique emails from both lists (Union).		
	3#	A cyber-security tool compares server logs. Find the common IP addresses between two lists (Intersection).		

WEEK NO		PROBLEMS WITH DESCRIPTION	PAGE NO.	SIGNATURE OF THE TEACHER WITH DATE
8	4#	A data visualization tool needs to plot a curve. Generate a list of tuples where the first element is a number and the second is five		
	5#	An NLP pre-processor needs to clean text. Use string methods to tokenize, normalize, find index, search for characters, and sanitize text.		
	6#	A data compression utility analyzes entropy. Calculate the frequency of every character present in a given string.		
	7#	A simple encryption tool obfuscates passwords. Write a program to reverse the order of characters in a string.		
	8#	A product labeling system generates "SKU codes". Generate a new string made of the first 2 and last 2 characters of a product name.		
	9#	A library catalog system contains sorted Book IDs. Implement Binary Search to find a specific Book ID.		
	10#	An educational algorithm visualizer needs to show sorting steps. Implement Insertion Sort and print the state of the list after every		
9 &	1#	A financial risk analysis tool needs to determine volatility. Identify the Maximum and Minimum values in a list of daily closing		
	2#	An inventory checking system allows a store manager to scan a product ID. Write a program to search for a specific element in a list of product IDs.		
	3#	A music playlist app needs to organize songs. Sort a raw list of song titles alphabetically.		
	4#	An election tally system has a dictionary of votes. Sort the dictionary to display the winner and the candidate with the fewest votes.		

WEEK NO		PROBLEMS WITH DESCRIPTION	PAGE NO.	SIGNATURE OF THE TEACHER WITH DATE
10	5#	A Data Science team is compiling a "Complete Patient Record". Write a script to concatenate three distinct dictionaries into one		
	6#	An embedded system needs to sort a small buffer. Implement the Optimized Bubble Sort algorithm (using a flag mechanism).		
	7#	A Big Data processing engine needs to sort a massive log file. Implement the Merge Sort algorithm using recursion.		
	8#	A real-time gaming engine needs a high-speed sorting algorithm. Implement the Quick Sort algorithm.		
11 & 12	1#	A DevOps engineer manages server configuration. Write a script to inspect, modify, clean up, and sort a dictionary of		
	2#	A linguistic research tool analyzes character distribution. Create a Histogram (dictionary) that counts how often each letter appears in a		
	3#	A corporate security system stores employee data. Write a program to perform a Reverse Lookup (find Key by Value) for a badge ID.		
	4#	An e-commerce pricing engine applies a "Flash Sale". Merge a Sale_Updates dictionary into a Catalog_Prices dictionary, overwriting existing values.		
	5#	A Math Learning App needs a lookup table. Generate a dictionary where keys are integers 1-15 and values are their squares.		
	6#	Implement the Matrix Chain Multiplication problem using dynamic programming.		
	7#	A GPS Navigation System calculates routes. Find the Shortest Path in a weighted graph using Dijkstra’s Algorithm.		

WEEK NO		PROBLEMS WITH DESCRIPTION	PAGE NO.	SIGNATURE OF THE TEACHER WITH DATE
13 & 14	1#	A government agency connects remote islands. Implement Kruskal's Algorithm to find the Minimum Spanning Tree (MST).		
	2#	A logistics company uses a delivery drone. Implement the Traveling Salesman Problem (TSP) to find the shortest route visiting all		
	3#	A museum security team places guards at intersections. Approximate a solution to the Vertex Cover Problem using a Greedy		

Week 2

1# Use interactive Shell to print:
• User said "Great app!"
• C:\new_folder\test\
Example 1: (*\n)*
• Input: User said "Great app!"\nC:\new_folder\test\
• Output:
User said "Great app!"
C:\new_folder\test\
Example 2: (*\t)*
• Input: User said\t"Great app!"
• Output:
User said "Great app!”

Code ->

```
➜  ~ python3
Python 3.12.2 (v3.12.2:6abddd9f6a, Feb  6 2024, 17:02:06) [Cla
Type "help", "copyright", "credits" or "license" for more info
>>> print("User said \"Great app!\"\nC:\\new_folder\\test\\")
User said "Great app!"
C:\new_folder\test\
>>> print("User said\t\"Great app!\"")
User said       "Great app!"
```

2# A shopkeeper uses Python shell for quick billing calculations.
Evaluate: 8+12, 94, 15-7, 18/3, 19/4, 19%4, 12%3, 37-5, 10-3*6-2
Example 1:
• Input: 10 / 0
• Output: ZeroDivisionError
Example 2:
• Input: 999999999999999 * 888888888888888
• Output: 888888888888887111111111111112

Code - >

```
>>> 8 + 12
20
>>> 9 ** 4
6561
>>> 15 - 7
8
>>> 18 / 3
6.0
>>> 19 / 4
4.75
>>>
>>> 19 % 4
3
>>> 12 % 3
0
>>> 37 - 5
32
>>> 10 - 3 * 6 - 2
-10
>>> -10
-10
```

3# A college form stores student names. Create variables firstName and lastName with your own names.

Code ->

```
>>> firstName = "Hasan"
>>> lastName = "Zaidi"
>>> print(firstName, lastName)
Hasan Zaidi
>>>
```

4# Print firstName and lastName in single line. (use f"")
(Try: print(firstName, lastName)

Code ->

```
>>> print(f"{firstName} {lastName}")
Hasan Zaidi
```

5# A heritage museum kiosk needs an automated display-and-billing script. Create a single Python file kioskSystem.py that performs the following

i) **Billing Module:**

The museum café calculates totals for simple arithmetic checks.
Write code to compute and print results for:

- 10 + 5
- 12 * 4
- 25 - 8
- 30 / 7
- 30 % 3

ii) **Information Display Module:**

After billing output, print two separate informational messages for visitors:

- "Welcome to the National Art Museum"
- "Digital exhibits were introduced in 2015"

Code ->

```
1   # Billing Module
2   print("Billing Calculations:")
3
4   print(10 + 5)       # Addition
5   print(12 * 4)       # Multiplication
6   print(25 - 8)       # Subtraction
7   print(30 / 7)       # Division
8   print(30 % 3)       # Modulus
9
10  # Information Display Module
11  print("\nVisitor Information:")
12  print("Welcome to the National Art Museum")
13  print("Digital exhibits were introduced in 2015")
```

Output ->

```
➜  Python LAB sem 6th python3 -u "/Users/xh
Billing Calculations:
15
48
17
4.285714285714286
0

Visitor Information:
Welcome to the National Art Museum
Digital exhibits were introduced in 2015
```

6# An engineer calculates load using a polynomial. Given a=3, b=-2, c=6, d=1,
e=4, f=7 and input x,

Compute: $ax^5 + bx^4 + cx^3 + dx^2 + ex + f^{1/2}$

Code ->

```
import math
a = 3
b = -2
c = 6
d = 1
e = 4
f = 7
x = float(input("Enter value of x: "))
result = (a * x**5 +
          b * x**4 +
          c * x**3 +
          d * x**2 +
          e * x +
          math.sqrt(f))
print("Result =", result)
```

Output ->

```
● → Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/week2/q6.py"
Enter value of x: 2
Result = 126.64575131106459
```

7# Write a program to compute the value of following algebraic expression-

The value of x and y will be read using input () function

$$\frac{1 + \frac{x}{y} + x^y}{2 + \frac{y}{x} + y^x}$$

Code ->

```
x = float(input("Enter value of x: "))
y = float(input("Enter value of y: "))

result = (1 + x / y + x * y) / (2 + y / x + y ** x)

print("Result =", result)
```

Output ->

```
● → Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/week2/q7.py"
Enter value of x: 23
Enter value of y: 3
Result = 8.249845356043418e-10
```

8# A technician optimizes multiplication using bitwise operations.
Read an integer and multiply it by 2 using <<.
Example 1:
• Input: n = 5
• Operation: n << 1
• Output: 10
Example 2:
• Input: n = 20
• Operation: n >> 1
• Output: 10

Code ->

```
1   n = int(input("Enter a number: "))
2
3   print("n << 1 =", n << 1)   # Multiply by 2
4   print("n >> 1 =", n >> 1)   # Divide by 2
```

Output ->

```
● ➜  Python LAB sem 6th python3 -u "/Use
Enter a number: 5
n << 1 = 10
n >> 1 = 2
```

Week Three

1# A bookstore’s billing app needs a quick utility script. Write a Python program
that reads N integers (prices) and computes:
i) SUM
ii) PRODUCT
iii) AVERAGE

Code - >

```python
n = int(input("Enter numbers of elemments:"))
sum = 0
product = 1

for i in range(1, n+1):
    num = int(input(f"Enter number {i}: "))
    sum += num;
    product *= num;

    average = sum/n;

print("Sum =", sum)
print("Product =", product)
print("Average =", average)
```

Output ->

```
→  Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/Week3/q1.py"
Enter numbers of elemments:3
Enter number 1: 123
Enter number 2: 321
Enter number 3: 100
Sum = 544
Product = 3948300
Average = 181.33333333333334
```

2# A weather station needs temperature conversion. Write a program to convert
Celsius to Fahrenheit. (Celsius values can be fractional.)

Code ->

```
celsius = float(input("Enter temperature in Celsius: "))
fahrenheit = (celsius *9/5) + 32
print("Temperature in Fahrenheit:", fahrenheit)
```

Output ->

```
➜  Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/Week3/q2.py"
Enter temperature in Celsius: 12
Temperature in Fahrenheit: 53.6
```

3# An architecture tool calculates dimensions. Write a program to compute the
Surface Area and Volume of a Cuboid given length, breadth, height.

Code->

```
1  l = float(input("Enter length: "))
2  b = float(input("Enter breadth: "))
3  h = float(input("Enter height: "))
4
5  surface_area = 2*(l*b + b*h + h*l)
6  volume = l*b*h
7
8  print("surface area = ", surface_area)
9  print("Volume = ", volume)
```

Output ->

```
➜  Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/Week3/q3.py"
Enter length: 23
Enter breadth: 34
Enter height: 45
surface area =  6694.0
Volume =  35190.0
```

4# Write a program to add two numbers without using arithmetic operators, using
bitwise operators.

Code ->

```
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))

while b!=0:
    carry = a&b
    a = a^b
    b = carry<<1

print("Sum = ", a)
```

Output->

```
● →  Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/Week3/q4.py"
Enter first number: 12
Enter second number: 34
Sum =  46
```

5# A hardware technician tests divider circuits. Read an integer and divide it by 4
using the bitwise >> operator.
Example 1:
• Input: n = 20
• Output: 5
Example 2:
• Input: n = -20
• Output: -5

Code ->

```
n = int(input("Enter an integer: "))
result = n>>2
print("Result after dividing by  4:", result)
```

Output ->

```
● →  Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/Week3/q5.py"
Enter an integer: 12
Result after dividing by  4: 3
```

6# A robotics controller swaps sensor IDs efficiently. Read two integers and exchange their values using XOR (^) operator.

Code ->

```
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))

a = a ^ b
b = a ^ b
a = a ^ b

print("After swapping")
print("a =", a)
print("b =", b)
```

Output ->

```
→  Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/Week3/q6.py"
Enter first number: 3
Enter second number: 9
After swapping
a = 9
b = 3
```

7# A gaming console optimizes score calculation. Multiply a number by 10 using
bitwise operators.

Code ->

```
n = int(input("Enter a number: "))
result = (n<<3) + (n<<1)
print("Result", result)
```

Output ->

```
→  Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/Week3/q7.py"
Enter a number: 12
Result 120
```

8# A data-logging device swaps configuration settings. Read two integers and
exchange their values using addition and subtraction (avoid using a third variable).

Code->

```
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))

a = a + b
b = a - b
a = a - b

print("After swapping")
print("a =", a)
print("b =", b)
```

Output -

```
➜  Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/Week3/q8.py"
Enter first number: 12
Enter second number: 23
After swapping
a = 23
b = 12
```

9# A biology simulation needs repeated multiplication. Write a program to compute the factorial of an integer using a loop.
Example 1:
• Input: 0
• Output: 1
Example 2:
• Input: -4
• Output: Invalid input

Code ->

```
n = int(input("Enter a number: "))
fact = 1
for i in range(1, n+1):
    fact *= i

print("Factorial =", fact)
```

Output ->

```
➜  Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/Week3/q9.py"
Enter a number: 3
Factorial = 6
```

10# A math module checks sequence validity. Write a program to determine whether a number is a Fibonacci number.

Code ->

```
n = int(input("Enter a number: "))
a, b= 0, 1
isfibo = False

while a<=n:
    if a ==n:
        isfibo = True
        break
    a, b = b, a+b

if isfibo:
    print("Fibonacci Number")
else:
    print("Not a fibonacci number")
```

Output ->

```
● → Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/Week3/q9.py"
Enter a number: 3
Factorial = 6
```

11# A security system verifies prime-number-based keys. Write a program to check whether an integer is prime, or else output its first factor.
Example 1:
• Input: 1
• Output: Not a prime number
Example 2:
• Input: -7
• Output: Invalid input

Code ->

```
n = int(input("Enter a number: "))

if n<= 1:
    print("Not Prime")
else:
    for i in range(2, n):
        if n%i == 0:
            print("Not prime, first factor:", i)
            break

    print("Prime number")
```

Output ->

```
● → Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/Week3/q11.py"
Enter a number: 7
Prime number
● → Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/Week3/q11.py"
Enter a number: 6
Not prime, first factor: 2
Prime number
```

12# A signal-processing unit finds harmony between two frequencies. Write a
program to compute the GCD of two numbers.
Example 1:
• Input: a = 0, b = 18
• Output: 18
Example 2:
• Input: a = 8, b = 15
• Output: 1

Code ->

```
1   a = int(input("Enter first number: "))
2   b = int(input("Enter second number: "))
3
4   while b != 0:
5       a, b = b, a%b
6
7   print("GCD =", a)
```

Output->

```
● → Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/Week3/q12.py"
Enter first number: 9
Enter second number: 3
GCD = 3
```

Week 4

1# A teacher wants to automate small mathematical tasks done repeatedly while
checking assignments. Define separate Python functions to perform the following for the teacher:

• Compute the average of any five marks obtained by a student.
• Convert a given temperature from Celsius to Fahrenheit for recording lab conditions.
• Calculate the perimeter of a rectangular notice board kept in the classroom using its length and width.

Example 1:

• Input: Marks = [-10, 0, 20, 30, 40]
• Output: Average = 16.0

Example 2:

• Input: Temperature = 36.5
• Output: 97.7°F

Code ->

```python
def calculate_average(marks):
    return sum(marks) / 5

def celsius_to_fahrenheit(celsius):
    return (celsius * 9/5) + 32

def rectangle_perimeter(length, width):
    return 2 * (length + width)

print("Average =", calculate_average([-10, 0, 20, 30, 40]))
print("Temperature =", celsius_to_fahrenheit(36.5), "°F")
print("Perimeter =", rectangle_perimeter(10, 5))
```

Output ->

```
● → Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/week 4/w4q1.py"
Average = 16.0
Temperature = 97.7 °F
Perimeter = 30
```

2# A science lab technician is calculating the volume of spherical containers used
for experiments. Write a Python function that accepts the radius and returns the volume of the sphere using the formula:

$$V = \frac{4}{3}\pi r^3$$

Find the volume when **radius = 6 cm**.
Example 1:
• Input: radius = 0.5
• Output: 0.52 cm³
Example 2:
• Input: radius = -3
• Output: Invalid radius

Code ->

```python
import math

# Function to calculate volume of a sphere
def volume_of_sphere(radius):
    if radius < 0:
        return "Invalid radius"
    return (4/3) * math.pi * radius**3

radius1 =float(input("Enter the radious :"))
result1 = volume_of_sphere(radius1)
print("Volume =", round(result1, 3), "cm³")
```

Output ->

```
→  Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/week 4/w4q2.py"
Enter the radious :3
Volume = 113.097 cm³
```

3# A landscaping company maintains circular flower beds, each with an outer
radius R and an inner unused radius r. The effective area used for fertilizer is
Effective Area= $\pi(R^2 - r^2)$

Write a Python function that takes R and r from the user, validates that R > r, and returns the effective usable area.

Example 1:

• Input: R=50.5, r=12.3

• Output: Effective Area = 7504.67

Example 2:

• Input: R=10, r=10

• Output: Invalid input

Code->

```python
import math

# Function to calculate effective area of circular flower bed
def effective_area(R, r):
    if R <= r:
        return "Invalid input"
    return math.pi * (R**2 - r**2)


R1 = float(input("Enter the outer radious :"))
r1 =float(input("Enter the inner radious :"))
area1 = effective_area(R1, r1)
print("Effective Area =", round(area1, 2))
```

Output ->

```
→  Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/week 4/w4q3.py"
Enter the outer radious :12
Enter the inner radious :4
Effective Area = 402.12
```

4# A data analyst needs factorial values to compute permutations.
Write a Python function to calculate the factorial of a non-negative integer.
Example 1:
• Input: 0
• Output: 1
Example 2:
• Input: 1
• Output: 1

Code->

```python
# Function to calculate factorial
def factorial(n):
    if n < 0:
        return "Invalid input"
    fact = 1
    for i in range(1, n + 1):
        fact = fact * i
    return fact

num1 = int(input("Enter the number:"))
print("Factorial =", factorial(num1))
```

Output ->

```
● → Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/week 4/w4q4.py"
Enter the number:1
Factorial = 1
● → Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/week 4/w4q4.py"
Enter the number:0
Factorial = 1
● → Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/week 4/w4q4.py"
Enter the number:5
Factorial = 120
```

5# An accountant wants a utility that extracts a digit-sum check value from transaction IDs. Write a Python function that returns the sum of digits of a given integer.
Example:
• Input: "0123"
• Output: 6
Code ->

```python
# Function to calculate sum of digits
def sum_of_digits(num):
    total = 0
    for digit in str(num):
        total += int(digit)
    return total

n = int(input("Enter the number:"))
result = sum_of_digits(n)
print("Sum of digits =", result)
```

Output->

```
● → Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/week 4/w4q5.py"
Enter the number:67
Sum of digits = 13
```

6# A math research assistant is verifying test values for numerical experiments.
Write a Python function to check whether a number is a perfect number. In number theory, a perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself. Example:6, 28 etc.
Example 1:
• Input: 1
• Output: Not a perfect number
Example 2:
• Input: 28
• Output: Perfect number

Code->

```python
number =int(input("Enter the number : "))
Tsum = 0
for i in range(1,number):
    if number%i==0:
        Tsum+=i

if Tsum ==number:
    print(number,"is the perfect number")
else:
    print(number, "is not the perfect number")
```

Output ->

```
● → Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/week 4/w4Q06.py"
Enter the number : 28
28 is the perfect number
```

7# Write a Python function to check whether a given number is a prime number
or not.
Example 1:
• Input: 0
• Output: Not prime
Example 2:
• Input: 1
• Output: Not prime

Code->

```python
def is_prime(n):
    if n <= 1:
        return False
    i = 2
    while i * i <= n:
        if n % i == 0:
            return False
        i += 1
    return True
num = int(input("Enter a number: "))
if is_prime(num):
    print("Prime number")
else:
    print("Not a prime number")
```

Output ->

```
➜  Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/week 4/w4Q07.py"
Enter a number: 2
Prime number
➜  Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/week 4/w4Q07.py"
Enter a number: 69
Not a prime number
```

8# A text-analysis tool needs statistics from a sentence. Write a Python function
that accepts a string and counts the number of uppercase and lowercase letters
using isupper(), islower(), upper(), lower().
Example 1:
• Input: "Hello@123"
• Output: Uppercase=1, Lowercase=4
Example 2:
• Input: "PYTHON"
• Output: Uppercase=6, Lowercase=0

Code ->

```python
def count_case(text):
    upper_count = 0
    lower_count = 0

    for ch in text:
        if ch.isupper():
            upper_count += 1
        elif ch.islower():
            lower_count += 1

    return upper_count, lower_count

text=str(input("Enter the string :"))
u1, l1 = count_case(text)
print("Uppercase =", u1, ", Lowercase =", l1)
```

Output ->

```
● → Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/week 4/w4q8.py"
Enter the string :XhXahdi
Uppercase = 2 , Lowercase = 5
```

Week 5

1# A data automation team is creating a utility module that performs multiple independent calculations based on user inputs. Write a Python program that accepts required inputs from the user and defines separate functions to perform:

• Find the largest of three numbers (positional arguments).
• Compute the volume based on user's chosen shape:
– cylinder (r, h), cube (a), or rectangular box (l, w, h).
• Compute the area of a rectangle.
• Compute the circumference of a circle.
• Exchange the values of two variables.
• Find the distance between two points (x1, y1) and (x2, y2) using math.dist().
(Use switch case to execute the required function)

Code->

```python
import math
# 1. Largest of three numbers (positional arguments)
def largest_of_three(a, b, c):
    return max(a, b, c)
# 2. Volume calculations
def volume_cylinder(r, h):
    return math.pi * r * r * h
def volume_cube(a):
    return a ** 3
def volume_rectangular_box(l, w, h):
    return l * w * h
# 3. Area of rectangle
def area_rectangle(l, w):
    return l * w
# 4. Circumference of circle
def circumference_circle(r):
    return 2 * math.pi * r
# 5. Exchange values
def swap_values(a, b):
    return b, a
# 6. Distance between two points
def distance_between_points(x1, y1, x2, y2):
    return math.dist((x1, y1), (x2, y2))

print("""
Choose an operation:
1. Largest of three numbers
2. Volume of a shape
3. Area of a rectangle
4. Circumference of a circle
5. Swap two values
6. Distance between two points
""")

choice = int(input("Enter your choice: "))

match choice:
    case 1:
        a, b, c = map(int, input("Enter three numbers: ").split())
        print("Largest =", largest_of_three(a, b, c))

    case 2:
        print("Choose shape: 1-Cylinder  2-Cube  3-Rectangular Box")
        shape = int(input("Enter shape choice: "))

        match shape:
            case 1:
                r, h = map(float, input("Enter radius and height: ").split())
                print("Volume =", volume_cylinder(r, h))
            case 2:
                a = float(input("Enter side: "))
                print("Volume =", volume_cube(a))
            case 3:
                l, w, h = map(float, input("Enter length width height: ").split())
                print("Volume =", volume_rectangular_box(l, w, h))
            case _:
                print("Invalid shape choice")

    case 3:
        l, w = map(float, input("Enter length and width: ").split())
        print("Area =", area_rectangle(l, w))

    case 4:
        r = float(input("Enter radius: "))
        print("Circumference =", circumference_circle(r))

    case 5:
        a, b = input("Enter two values: ").split()
        a, b = swap_values(a, b)
        print("After swap:", a, b)

    case 6:
        x1, y1 = map(float, input("Enter x1 y1: ").split())
        x2, y2 = map(float, input("Enter x2 y2: ").split())
        print("Distance =", distance_between_points(x1, y1, x2, y2))

    case _:
        print("Invalid choice")
```

Output ->

```
● → Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/week 5/q1.py"

Choose an operation:
1. Largest of three numbers
2. Volume of a shape
3. Area of a rectangle
4. Circumference of a circle
5. Swap two values
6. Distance between two points

Enter your choice: 3
Enter length and width:  12 23
Area = 276.0
```

2# A billing system receives an unknown count of numeric entries per invoice.
Write a Python function that accepts arbitrary integers and returns their sum. (Hint: use *args)
Example:
• Input: ()
• Output: 0
Code ->

```python
def total_sum(*args):
    return sum(args)

print(total_sum())
print(total_sum(10,20,5))
```

Output ->

```
● ➜  Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/week 5/q2.py"
0
35
```

3# A text-filtering tool needs to remove noisy characters occurring at odd positions. Write a Python program that takes a string and removes all characters at even index values.
Example 1:
• Input: "H@e#l$l"
• Output: "el"
Example 2:
• Input: ""
• Output: ""

Code ->

```python
s = input("enter the string:")

# removing noisy(special) character at odd place ->
step1= ""
for i in range(len(s)):
    if i%2!=0 and not s[i].isalnum():
        continue
    step1 +=s[i]

#removing the char at even place->
final=""
for i in range(len(step1)):
    if i%2!=0:
        final+=step1[i]

print(final)
```

Output ->

```
● ➜  Python LAB sem 6th python3 -u "
enter the string:hasan
aa
```

4# Write a Python script that takes input from the user and displays that input back in upper and lower cases.

Code->

```
s =input("enter a string: ")
print("upper case: ", s.upper())
print("lower case: ", s.lower())
```

Output ->

```
Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/week 5/q4.py"
enter a string: hasan
upper case:  HASAN
lower case:  hasan
```

5# A simple analytics module must calculate keyword frequency in user feedback. Write a Python program to count occurrences of each word in a sentence.

Example 1:

• Input: "Hello hello world"

• Output: {'hello': 2, 'world': 1}

Example 2:

• Input: "Hi, Hi!"

• Output: {'hi': 2}

Code->

```
import string
sentence = input()
# Convert to lowercase
sentence = sentence.lower()
# Remove punctuation
for ch in string.punctuation:
    sentence = sentence.replace(ch, "")
# Split into words
words = sentence.split()
# Count occurrences
word_count = {}
for word in words:
    if word in word_count:
        word_count[word] += 1
    else:
        word_count[word] = 1
print(word_count)
```

Output ->

```
Python LAB sem 6th python3 -u "/Users/xhxaidi/Python LAB sem 6th/week 5/q5.py"
Hello hello world
{'hello': 2, 'world': 1}
```