Week-1

1. Write a Python program to check whether a given number is even or odd.

Algorithm:

1.Start

2.Read an integer number n from the user

3.Check if n % 2 == 0

4.If true, print "Even number"

5.Else, print "Odd number"

6.Stop

```python
# Program to check whether a number is Even or Odd

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

if n % 2 == 0:
    print("The number is Even")
else:
    print("The number is Odd")
```

```
Enter a number:  6

The number is Even
```

1. Write a Python program to check whether a number is positive, negative, or zero.

Algorithm:

1.Start

2.Read a number n from the user

3.If n > 0, print "Positive number"

4.Else if n < 0, print "Negative number"

5.Else, print "Zero"

6.Stop

```python
# Program to check whether a number is Positive, Negative, or Zero

n = float(input("Enter a number: "))

if n > 0:
    print("The number is Positive")
```

```
elif n < 0:
    print("The number is Negative")
else:
    print("The number is Zero")
```

```
Enter a number:  4
```

```
The number is Positive
```

1. Write a Python program to find the largest among three numbers.

Algorithm:

1.Start

2.Read three numbers a, b, and c from the user

3.If a ≥ b and a ≥ c, print a is the largest

4.Else if b ≥ a and b ≥ c, print b is the largest

5.Else, print c is the largest

6.Stop

```
# Program to find the largest among three numbers

a = float(input("Enter first number: "))
b = float(input("Enter second number: "))
c = float(input("Enter third number: "))

if a >= b and a >= c:
    print("The largest number is:", a)
elif b >= a and b >= c:
    print("The largest number is:", b)
else:
    print("The largest number is:", c)
```

```
Enter first number:  55
Enter second number:  42
Enter third number:  67
```

```
The largest number is: 67.0
```

1. Write a Python program to check whether a given number is a prime number.

Algorithm:

1.Start

2.Read an integer number n from the user

3.If n ≤ 1, print "Not a Prime number" and go to step 7

4.Set a variable flag = 0

5.For i from 2 to n−1:

If n % i == 0, set flag = 1 and break

6.If flag == 0, print "Prime number"

7.Else, print "Not a Prime number"

8.Stop

```python
# Program to check whether a number is Prime or not

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

if n <= 1:
    print("Not a Prime number")
else:
    flag = 0
    for i in range(2, n):
        if n % i == 0:
            flag = 1
            break

    if flag == 0:
        print("Prime number")
    else:
        print("Not a Prime number")
```

```
Enter a number:  7

Prime number
```

Week-2

1. Write a Python program to find the factorial of a number.

Algorithm:

1.Start

2.Read an integer number n from the user

3.If n < 0, print "Factorial not defined for negative numbers"

4.Else

Initialize fact = 1

For i from 1 to n, do fact = fact × i

5.Print the value of fact

6.Stop

```python
# Program to find the factorial of a number

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

if n < 0:
    print("Factorial not defined for negative numbers")
else:
    fact = 1
    for i in range(1, n + 1):
        fact = fact * i
    print("Factorial of", n, "is:", fact)
```

```
Enter a number:  5

Factorial of 5 is: 120
```

1. Write a Python program to check whether a number is a palindrome.

Algorithm:

1.Start

2.Read an integer number n from the user

3.Store the value of n in a temporary variable temp

4.Initialize rev = 0

5.While n > 0:

digit = n % 10

rev = rev × 10 + digit

n = n // 10

6.If temp == rev, print "Palindrome number"

7.Else, print "Not a Palindrome number"

8.Stop

```python
# Program to check whether a number is Palindrome or not

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

temp = n
rev = 0

while n > 0:
    digit = n % 10
    rev = rev * 10 + digit
    n = n // 10

if temp == rev:
    print("Palindrome number")
else:
    print("Not a Palindrome number")
```

```
Enter a number:  6

Palindrome number
```

1. Write a Python program to check whether a given string is a palindrome.

Algorithm:

1.Start

2.Read a string s from the user

3.Reverse the string and store it in rev

4.Compare the original string s with the reversed string rev

5.If both are equal, print "Palindrome string"

6.Else, print "Not a Palindrome string"

7.Stop

```
# Program to check whether a string is Palindrome or not

s = input("Enter a string: ")

rev = s[::-1]

if s == rev:
    print("Palindrome string")
else:
    print("Not a Palindrome string")
```

```
Enter a string:  Hello

Not a Palindrome string
```

Week-3

1. Write a Python program to print the Fibonacci series up to N terms.

Algorithm:

1.Start

2.Read an integer n (number of terms) from the user

```
3.Initialize a = 0, b = 1
4.Print a and b
5.Repeat for i from 3 to n:
c = a + b
Print c
Assign a = b, b = c
6.Stop
```

```python
# Program to print Fibonacci series up to N terms

n = int(input("Enter number of terms: "))

a, b = 0, 1

if n <= 0:
    print("Please enter a positive number")
elif n == 1:
    print(a)
else:
    print("Fibonacci series:")
    print(a, b, end=" ")
    for i in range(3, n + 1):
        c = a + b
        print(c, end=" ")
        a = b
        b = c
```

```
Enter number of terms:  7

Fibonacci series:
0 1 1 2 3 5 8
```

1. Write a Python program to find the sum of digits of a number.

Algorithm:

```
1.Start
2.Read an integer number n from the user
3.Initialize sum = 0
4.While n > 0:
digit = n % 10
sum = sum + digit
```

n = n // 10

5.Print the value of sum

6.Stop

```
# Program to find the sum of digits of a number

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

sum_digits = 0

while n > 0:
    digit = n % 10
    sum_digits = sum_digits + digit
    n = n // 10

print("Sum of digits is:", sum_digits)

Enter a number:  22

Sum of digits is: 4
```

1. Write a Python program to count vowels and consonants in a string.

Algorithm:

1.Start

2.Read a string s from the user

3.Convert the string to lowercase

4.Initialize vowels = 0 and consonants = 0

5.For each character in the string:

If the character is an alphabet:

```
If it is a vowel (a, e, i, o, u), increment vowels

Else, increment consonants
```

6.Print the number of vowels and consonants

7.Stop

```
# Program to count vowels and consonants in a string

s = input("Enter a string: ")
s = s.lower()

vowels = 0
```

```
consonants = 0

for ch in s:
    if ch.isalpha():
        if ch in 'aeiou':
            vowels += 1
        else:
            consonants += 1

print("Number of vowels:", vowels)
print("Number of consonants:", consonants)

Enter a string:  Good Morning

Number of vowels: 4
Number of consonants: 7
```

Week-4

1. Write a Python program to reverse a string without using built-in functions.

Algorithm:

```
1.Start
2.Read a string s from the user
3.Initialize an empty string rev = ""
4.For each character ch in the string s:
Add ch in front of rev (rev = ch + rev)
5.Print the reversed string rev
6.Stop
```

```
# Program to reverse a string without using built-in functions

s = input("Enter a string: ")

rev = ""

for ch in s:
    rev = ch + rev

print("Reversed string is:", rev)

Enter a string:  Good Morning

Reversed string is: gninroM dooG
```

1. Write a Python program to count the occurrence of each character in a string.

Algorithm:

1.Start

2.Read a string s from the user

3.Initialize an empty dictionary count = {}

4.For each character ch in the string:

If ch is already in count, increment its value by 1

Else, add ch to count with value 1

5.Display each character and its occurrence

6.Stop

```python
# Program to count the occurrence of each character in a string

s = input("Enter a string: ")

count = {}

for ch in s:
    if ch in count:
        count[ch] += 1
    else:
        count[ch] = 1

print("Character occurrences:")
for ch in count:
    print(ch, ":", count[ch])
```

```
Enter a string:  Hello World

Character occurrences:
H : 1
e : 1
l : 3
o : 2
  : 1
W : 1
r : 1
d : 1
```

1. Write a Python program to create a simple calculator using conditional statements.

Algorithm:

1.Start

2.Display the calculator menu:

Addition

Subtraction

Multiplication

Division

3.Read the user’s choice

4.Read two numbers a and b

5.Use conditional statements to perform the selected operation:

If choice is 1, add the numbers

If choice is 2, subtract the numbers

If choice is 3, multiply the numbers

If choice is 4, divide the numbers (check division by zero)

6.Display the result

7.Stop

```python
# Simple Calculator using Conditional Statements

print("Simple Calculator")
print("1. Addition")
print("2. Subtraction")
print("3. Multiplication")
print("4. Division")

choice = int(input("Enter your choice (1-4): "))

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

if choice == 1:
    print("Result:", a + b)
elif choice == 2:
    print("Result:", a - b)
elif choice == 3:
    print("Result:", a * b)
elif choice == 4:
    if b != 0:
        print("Result:", a / b)
    else:
        print("Division by zero is not allowed")
else:
    print("Invalid choice")
```

```
Simple Calculator
1. Addition
2. Subtraction
3. Multiplication
4. Division

Enter your choice (1-4):  1
Enter first number:  5
Enter second number:  4

Result: 9.0
```

Week-5

1. Write a Python program to implement a menu-driven calculator using a loop (repeat until the user exits).

Algorithm:

1.Start

2.Repeat the following steps until the user chooses Exit:

1.Display the calculator menu:

Addition

Subtraction

Multiplication

Division

Exit

2.Read the user’s choice

3.If the choice is Exit, break the loop

4.Else, read two numbers a and b

5.Use conditional statements to perform the selected operation

Addition → a + b

Subtraction → a - b

Multiplication → a * b

Division → check division by zero

6.Display the result

3.Stop

```python
# Menu-driven Calculator using Loop

while True:
    print("\nMenu:")
    print("1. Addition")
    print("2. Subtraction")
    print("3. Multiplication")
    print("4. Division")
    print("5. Exit")

    choice = int(input("Enter your choice (1-5): "))

    if choice == 5:
        print("Exiting calculator...")
        break

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

    if choice == 1:
        print("Result:", a + b)
    elif choice == 2:
        print("Result:", a - b)
    elif choice == 3:
        print("Result:", a * b)
    elif choice == 4:
        if b != 0:
            print("Result:", a / b)
        else:
            print("Division by zero is not allowed")
    else:
        print("Invalid choice. Please try again.")
```

```
Menu:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit

Enter your choice (1-5):  1
Enter first number:  6
Enter second number:  4

Result: 10.0

Menu:
1. Addition
2. Subtraction
3. Multiplication
```

```
4. Division
5. Exit

Enter your choice (1-5):  2
Enter first number:  5
Enter second number:  3

Result: 2.0

Menu:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit

Enter your choice (1-5):  3
Enter first number:  2
Enter second number:  3

Result: 6.0

Menu:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit

Enter your choice (1-5):  4
Enter first number:  6
Enter second number:  3

Result: 2.0

Menu:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit

Enter your choice (1-5):  5

Exiting calculator...
```

1. Write a Python program to generate a multiplication table for a given number (loop until the user stops).

Algorithm:

1.Start

```
2.Repeat the following steps until the user chooses to stop:
    1.Read an integer number n from the user
    2.For i from 1 to 10:
        Print n × i = result
    3.Ask the user whether they want to continue (Y/N)
    4.If the user enters N or n, exit the loop
3.Stop
```

```python
# Program to generate multiplication table until user stops

while True:
    n = int(input("Enter a number: "))

    print("Multiplication Table of", n)
    for i in range(1, 11):
        print(n, "x", i, "=", n * i)

    choice = input("Do you want to continue? (Y/N): ")

    if choice.lower() == 'n':
        print("Program terminated.")
        break
```

```
Enter a number:  3

Multiplication Table of 3
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30

Do you want to continue? (Y/N):  N

Program terminated.
```

1. Write a Python program to print different patterns using loop concepts (e.g., star patterns, number patterns).

Algorithm:

1.Start

2.Read an integer n (number of rows) from the user

3.Print Star Pattern using nested loops

4.Print Number Pattern using nested loops

5.Stop

```
# Program to print different patterns using loops

n = int(input("Enter number of rows: "))

print("\nStar Pattern:")
for i in range(1, n + 1):
    for j in range(i):
        print("*", end=" ")
    print()

print("\nNumber Pattern:")
for i in range(1, n + 1):
    for j in range(1, i + 1):
        print(j, end=" ")
    print()
```

```
Enter number of rows:  5


Star Pattern:
*
* *
* * *
* * * *
* * * * *

Number Pattern:
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Week-6

1. Write a Python function that takes a user’s name and prints a greeting message.

Algorithm:

1.Start

2.Define a function greet(name)

3.Inside the function, print a greeting message using the given name

4.Read the user’s name from input

5.Call the function with the user’s name as argument

6.Stop

```
# Function to print a greeting message

def greet(name):
    print("Hello", name + "! Welcome to Python programming.")

# Taking input from user
user_name = input("Enter your name: ")

# Calling the function
greet(user_name)
```

```
Enter your name:  Python

Hello Python! Welcome to Python programming.
```

1. Write a Python function that accepts two numbers and returns their sum.

Algorithm:

1.Start

2.Define a function add(a, b)

3.Inside the function, calculate sum = a + b

4.Return the value of sum

5.Read two numbers from the user

6.Call the function with the given numbers

7.Display the returned result

8.Stop

```
# Function to return the sum of two numbers

def add(a, b):
    return a + b

# Taking input from user
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))

# Function call
result = add(num1, num2)
```

```
print("Sum of the two numbers is:", result)
```

```
Enter first number:  9
Enter second number:  3
```

```
Sum of the two numbers is: 12.0
```

Week-7

1. Write a Python recursive function to find the factorial of a number.

Algorithm:

1.Start

2.Define a recursive function factorial(n)

3.If n == 0 or n == 1, return 1 (base case)

4.Else, return n × factorial(n − 1) (recursive case)

5.Read an integer number n from the user

6.Call the function and display the result

7.Stop

```
# Recursive function to find factorial of a number

def factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n - 1)

# Taking input from user
n = int(input("Enter a number: "))

if n < 0:
    print("Factorial is not defined for negative numbers")
else:
    result = factorial(n)
    print("Factorial of", n, "is:", result)
```

```
Enter a number:  3
```

```
Factorial of 3 is: 6
```

1. Write a Python lambda function to check whether a number is even.

Algorithm:

1.Start

2.Define a lambda function that takes a number n

3.Check the condition n % 2 == 0

4.Return True if the condition is satisfied, else False

5.Read a number from the user

6.Call the lambda function and display the result

7.Stop

```python
# Lambda function to check whether a number is even

is_even = lambda n: n % 2 == 0

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

if is_even(n):
    print("The number is Even")
else:
    print("The number is Odd")
```

```
Enter a number:  5

The number is Odd
```

1. Write a Python program to calculate factorial using recursion with input validation.

Algorithm:

1.Start

2.Define a recursive function factorial(n)

3.If n == 0 or n == 1, return 1 (base case)

4.Else, return n × factorial(n − 1)

5.Read an integer number n from the user

6.Validate input:

If n < 0, print "Invalid input"

Else, call the recursive function

7.Display the factorial result

8.Stop

```python
# Factorial using recursion with input validation

```

```python
def factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n - 1)

# Taking input from user
n = int(input("Enter a number: "))

# Input validation
if n < 0:
    print("Invalid input! Factorial is not defined for negative numbers.")
else:
    result = factorial(n)
    print("Factorial of", n, "is:", result)
```

```
Enter a number:  5

Factorial of 5 is: 120
```

Week-8

1. Write a Python program to create a Library Book Management System using functions.

Algorithm:

1.Start

2.Create an empty list to store book names

3.Define the following functions:

add_book() → Add a book to the library

remove_book() → Remove a book from the library

display_books() → Display all available books

4.Display a menu with options:

Add Book

Remove Book

Display Books

Exit

5.Use a loop to repeat the menu until the user chooses Exit

6.Call the appropriate function based on user choice

7.Stop

```python
# Library Book Management System using Functions

library = []

def add_book():
    book = input("Enter book name to add: ")
    library.append(book)
    print("Book added successfully.")

def remove_book():
    book = input("Enter book name to remove: ")
    if book in library:
        library.remove(book)
        print("Book removed successfully.")
    else:
        print("Book not found.")

def display_books():
    if not library:
        print("Library is empty.")
    else:
        print("Books available in library:")
        for book in library:
            print("-", book)

while True:
    print("\nLibrary Menu")
    print("1. Add Book")
    print("2. Remove Book")
    print("3. Display Books")
    print("4. Exit")

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

    if choice == 1:
        add_book()
    elif choice == 2:
        remove_book()
    elif choice == 3:
        display_books()
    elif choice == 4:
        print("Exiting Library System...")
        break
    else:
        print("Invalid choice. Please try again.")
```

```
Library Menu
1. Add Book
2. Remove Book
```

```
3. Display Books
4. Exit

Enter your choice:  1
Enter book name to add:  Smile

Book added successfully.

Library Menu
1. Add Book
2. Remove Book
3. Display Books
4. Exit

Enter your choice:  2
Enter book name to remove:  Smile

Book removed successfully.

Library Menu
1. Add Book
2. Remove Book
3. Display Books
4. Exit

Enter your choice:  3

Library is empty.

Library Menu
1. Add Book
2. Remove Book
3. Display Books
4. Exit

Enter your choice:  4

Exiting Library System...
```

Week-9

1. Write a Python project to build a Calculator using modular programming (separate module for operations).

Folder Structure:

```
Calculator_Project/
├── operations.py
└── main.py
```

Calculator_Project/ ├── operations.py └── main.py

```python
# operations module (functions)

def add(a, b):
    return a + b

def subtract(a, b):
    return a - b

def multiply(a, b):
    return a * b

def divide(a, b):
    if b != 0:
        return a / b
    else:
        return "Division by zero is not allowed"


# main program (menu-driven calculator)

while True:
    print("\nCalculator Menu")
    print("1. Addition")
    print("2. Subtraction")
    print("3. Multiplication")
    print("4. Division")
    print("5. Exit")

    choice = int(input("Enter your choice (1-5): "))

    if choice == 5:
        print("Exiting Calculator...")
        break
    a = float(input("Enter first number: "))
    b = float(input("Enter second number: "))

    if choice == 1:
        print("Result:", add(a, b))
    elif choice == 2:
        print("Result:", subtract(a, b))
    elif choice == 3:
        print("Result:", multiply(a, b))
    elif choice == 4:
        print("Result:", divide(a, b))
    else:
        print("Invalid choice")
```

```
Calculator Menu
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit

Enter your choice (1-5):  1
Enter first number:  5
Enter second number:  4

Result: 9.0

Calculator Menu
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit

Enter your choice (1-5):  2
Enter first number:  6
Enter second number:  4

Result: 2.0

Calculator Menu
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit

Enter your choice (1-5):  3
Enter first number:  7
Enter second number:  5
```

```
Result: 35.0

Calculator Menu
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit

Enter your choice (1-5):  4
Enter first number:  8
Enter second number:  4

Result: 2.0

Calculator Menu
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit

Enter your choice (1-5):  5

Exiting Calculator...
```

Week-10

1. Write a Python program that applies modular programming principles and defines multiple reusable functions.

Folder Structure

Utility_Project/ ├── utilities.py └── main.py

```python
# Reusable functions (module part)

def is_even(n):
    return n % 2 == 0

def factorial(n):
    fact = 1
    for i in range(1, n + 1):
        fact *= i
    return fact

def sum_of_digits(n):
    total = 0
    while n > 0:
        total += n % 10
        n //= 10
    return total


# Main program

while True:
    print("\nMenu")
    print("1. Check Even or Odd")
    print("2. Find Factorial")
    print("3. Find Sum of Digits")
    print("4. Exit")

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

    if choice == 4:
        print("Exiting program...")
        break

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

    if choice == 1:
        print("Even" if is_even(n) else "Odd")
    elif choice == 2:
        print("Factorial:", factorial(n))
    elif choice == 3:
        print("Sum of digits:", sum_of_digits(n))
    else:
        print("Invalid choice")
```

```
Menu
1. Check Even or Odd
2. Find Factorial
3. Find Sum of Digits
4. Exit

Enter your choice:  1
Enter a number:  5

Odd

Menu
1. Check Even or Odd
2. Find Factorial
3. Find Sum of Digits
4. Exit

Enter your choice:  2
Enter a number:  5
```

```
Factorial: 120

Menu
1. Check Even or Odd
2. Find Factorial
3. Find Sum of Digits
4. Exit

Enter your choice:  3
Enter a number:  33

Sum of digits: 6

Menu
1. Check Even or Odd
2. Find Factorial
3. Find Sum of Digits
4. Exit

Enter your choice:  4

Exiting program...
```

Week-11

1. Write a Python program using modular programming principles and demonstrate:

• Input validation

• Testing (minimum 3 test cases)

• Debugging practice with comments

Folder Structure

Validation_Project/ ├── operations.py └── main.py

```
# operations module (reusable functions)

def add(a, b):
    return a + b

def subtract(a, b):
    return a - b

def is_valid_number(value):
    try:
        float(value)
        return True
    except ValueError:
        return False
```

```python
# main program

def get_number(message):
    while True:
        value = input(message)

        # Debugging comment:
        # If input is not valid, ask again
        if is_valid_number(value):
            return float(value)
        else:
            print("Invalid input! Please enter a valid number.")


print("Simple Calculator with Validation")

a = get_number("Enter first number: ")
b = get_number("Enter second number: ")

print("Addition:", add(a, b))
print("Subtraction:", subtract(a, b))
```

```
Simple Calculator with Validation

Enter first number:  6
Enter second number:  3

Addition: 9.0
Subtraction: 3.0
```

Week-12

1. Write a Python project for a User Registration System with input validation, testing, and debugging documentation.

Folder Structure

User_Registration_Project/ ├── validation.py └── main.py

```python
# Validation functions (module part)

def validate_username(username):
    return len(username) >= 3

def validate_password(password):
    return len(password) >= 6

def validate_age(age):
    return age.isdigit() and int(age) >= 18
```

```python
# Main program

def register_user():

    while True:
        username = input("Enter username: ")

        # Debugging: username validation
        if validate_username(username):
            break
        else:
            print("Invalid username! Must be at least 3 characters.")

    while True:
        password = input("Enter password: ")

        # Debugging: password validation
        if validate_password(password):
            break
        else:
            print("Invalid password! Must be at least 6 characters.")

    while True:
        age = input("Enter age: ")

        # Debugging: age validation
        if validate_age(age):
            break
        else:
            print("Invalid age! Must be 18 or above.")

    print("\nUser Registered Successfully!")
    print("Username:", username)
    print("Age:", age)


# Run the registration system
register_user()
```

```
Enter username:  Python
Enter password:  Python@123
Enter age:  30


User Registered Successfully!
Username: Python
Age: 30
```