IoT – Python Assignment – 03

----

1. String Built-in Methods

Theory:

Python strings provide many built-in methods to manipulate text.

Code:

```
s = "  Hello IoT World  "

print("Original:", s)
print("Upper:", s.upper())
print("Lower:", s.lower())
print("Title:", s.title())
print("Strip:", s.strip())
print("Replace:", s.replace("IoT", "Python"))
print("Find 'World':", s.find("World"))
print("Count 'l':", s.count("l"))
print("Split:", s.split())
```

----

2. String Slicing

Theory:

String slicing extracts part of a string using index positions.

Code:

```
s = input("Enter a string: ")
print("First 5 characters:", s[:5])
print("Last 5 characters:", s[-5:])
print("Reverse:", s[::-1])
```

----

3. Vowels, Consonants and Ratio

Theory:

Functions are used to count vowels and consonants.

Code:

```
def count_vowels(s):
    vowels = "aeiouAEIOU"
    return sum(1 for ch in s if ch in vowels)

def count_consonants(s):
    return sum(1 for ch in s if ch.isalpha() and ch not in "aeiouAEIOU")

s = input("Enter string: ")
v = count_vowels(s)
c = count_consonants(s)
print("Vowels:", v)
print("Consonants:", c)
if c != 0:
    print("Ratio:", v/c)
```

----

4. Menu Driven Calculator

Theory:

Menu driven programs allow user to select operations.

Code:

```
def add(a, b): return a + b
def sub(a, b): return a - b
def mul(a, b): return a * b
def div(a, b): return "Error" if b == 0 else a / b

while True:
    print("1.Add 2.Sub 3.Mul 4.Div 5.Exit")
    ch = int(input("Enter choice: "))
    if ch == 5:
        break
    a = float(input("Enter a: "))
    b = float(input("Enter b: "))

    if ch == 1: print(add(a, b))
    elif ch == 2: print(sub(a, b))
    elif ch == 3: print(mul(a, b))
    elif ch == 4: print(div(a, b))
```

----

5. Default Arguments & Function as Argument

```
def greet(name="User"):
    print("Hello", name)

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

def apply(a, b, func):
    return func(a, b)

greet()
greet("Prathmesh")
print(apply(10, 20, add))
```

----

6. Calculate Function

```
def add(a, b): return a + b
def sub(a, b): return a - b

def calculate(a, b, func):
    return func(a, b)

print(calculate(5, 3, add))
print(calculate(5, 3, sub))
```

----

7. Recursion: Factorial and Power

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

def power(a, b):
    if b == 0:
        return 1
    return a * power(a, b-1)

print("Factorial:", factorial(5))
print("Power:", power(2, 4))
```