	SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGG.	**LABORATORY MANUAL**
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE : Write python program to print list of numbers using range and for loop.	
EXPERIMENT NO. : **SSGMCE/WI/IT/01/3IT09/01**	ISSUE NO. : 00	ISSUE DATE : 30.07.2023
REV. DATE :	REV. NO. :	DEPTT. : INFORMATION TECHNOLOGY
LABORATORY : 3IT09 COMPUTER SKILL LAB – I	SEMESTER : III	PAGE: 1 OF 2

1.0) AIM: Write python program to print list of numbers using range and for loop

2.0) SCOPE: This Python program is designed to introduce students to fundamental programming concepts, specifically utilizing loops and the **range** function to print a list of numbers. It offers a practical exercise that promotes comprehension of loops and data

3.0) FACILITIES/ APPARATUS:

1. Python development environment (e.g., IDLE)
2. Input mechanism (keyboard)
3. Computer with Python installed

4.0) THEORY:

The program demonstrates the implementation of the **range** function and the **for** loop in Python to generate and print a list of numbers.

4.1) Introduction about Range () and Looping Statements

Range: The **range()** function generates a sequence of numbers that can be used for iteration. It is commonly used with looping statements to perform repetitive tasks.

Example:

```
# Using range to create a sequence of numbers
for num in range(5):
    print(num)
```

Output:

```
0
1
2
3
4
```

Different Looping Statements:

1. **For Loop:** The **for** loop is used to iterate over a sequence (such as a list, tuple, or string) and execute a block of code for each item in the sequence.

Example:

```
fruits = ["apple", "banana", "orange"]
for fruit in fruits:
    print(fruit)
```

Output:

```
apple
banana
orange
```

2. **While Loop:** The **while** loop repeatedly executes a block of code as long as a condition is true.

Example:

```
count = 0
while count < 3:
    print("Count:", count)
    count += 1
```

Output:

```
Count: 0
Count: 1
Count: 2
```

3. **Nested Loops:** Nested loops involve placing one loop inside another. This is useful for iterating through multidimensional data structures.

Example:

```python
rows = 3
columns = 4
for i in range(rows):
    for j in range(columns):
        print(f"Row: {i}, Column: {j}")
```

Output:

```
Row: 0, Column: 0
Row: 0, Column: 1
Row: 0, Column: 2
Row: 0, Column: 3
Row: 1, Column: 0
Row: 1, Column: 1
Row: 1, Column: 2
Row: 1, Column: 3
Row: 2, Column: 0
Row: 2, Column: 1
Row: 2, Column: 2
Row: 2, Column: 3
```

4. **Break and Continue Statements:** The break statement is used to exit a loop prematurely, while the continue statement skips the current iteration and moves to the next one.

Example:

```python
for num in range(1, 6):
    if num == 3:
        break
    print(num)
```

Output:

```
1
2
```

Python's range() function, along with for and while loops, allows programmers to efficiently perform repetitive tasks and iterate over sequences of data. Nested loops and control statements like break and continue enhance the flexibility and power of Python's looping mechanisms. Understanding these concepts is crucial for building structured and efficient programs.

The following steps detail the program's execution:

4.2) Program Explanation:

1. **Range Definition:** Begin by defining a range of numbers using the **start** and **end** values. These values determine the range boundaries, where **start** is the lower limit, and **end** is the upper limit. For instance, setting **start = 1** and **end = 10** defines a range from 1 to 10.
2. **For Loop:** The **for** loop is used to iterate through the sequence of numbers generated by the **range** function. The loop iterates through the numbers inclusively from **start** to **end**, and for each iteration, a variable (e.g., **num**) is assigned the current value.
3. **Print Function:** Within the loop, the **print** function is employed to display the value of the **num** variable. This results in each number within the range being printed on a separate line.

Example Syntax and Description:

```
start = 1
end = 10

print("List of numbers:")
for num in range(start, end + 1):
    print(num)
```

4.3) Explanation of Execution:

1. The range is defined from 1 to 10.

2. The **for** loop iterates through each number in the range (1 to 10).
3. The **print** function is used to display each number, resulting in the complete list of numbers being printed.

5.0) Conclusion:

This programming experiment offers a hands-on opportunity for students to grasp the essence of loops and the **range** function in Python. By practicing generating and displaying a sequence of numbers, students gain practical insight into loop constructs and data manipulation. The learning outcomes encompass fundamental programming skills, an understanding of iteration, and the ability to apply these concepts to broader coding scenarios.