	SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGG.	**LABORATORY MANUAL**
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE : Write python program to store strings in list and then print them.	
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 store strings in list and then print them.

2.0) SCOPE: This Python program is aimed at introducing students to the concept of storing strings in lists and subsequently printing them. It offers a hands-on exercise to familiarize students with data manipulation using lists and basic input-output operations.

3.0) FACILITIES/ APPARATUS:

1. Python development environment (e.g., IDLE)
2. Input mechanism (keyboard)

4.0) THEORY:

The program illustrates the process of storing strings in a list and then printing them. The following steps provide a comprehensive understanding of the program's execution:

4.1) Strings in Python:

In Python, a string is a sequence of characters enclosed within single, double, or triple quotes. Strings are versatile data types used to represent text and can include letters, numbers, symbols, and spaces.

Example:

```
message = "Hello, World!"
print(message)
```

Output:

```
Hello, World!
```

Storing Strings in a List and Printing Them:

A list can store multiple strings, making it useful for managing collections of text data.

Example:

```python
# Storing strings in a list
fruits = ["apple", "banana", "orange", "grape"]

# Printing each fruit in the list
for fruit in fruits:
    print(fruit)
```

Output:

```
apple
banana
orange
grape
```

Explanation:

1. We define a list named fruits that contains strings representing different fruits.
2. Using a for loop, we iterate through the fruits list.
3. The loop prints each fruit's name using the print() function.

Strings in Python are fundamental data types used to represent text, while lists are versatile data structures for storing collections of items, including strings. By combining strings within lists and using loops, programmers can effectively manage and manipulate textual data in various ways.

4.2) Program Explanation:

1. **List Creation:** Begin by initializing an empty list named **string_list**.
2. **User Input:** Prompt the user to input the number of strings they intend to store. Using a **for** loop, iterate through the specified number to gather each string.
3. **Appending Strings:** Utilize the **append** function to add the user's input (strings) to the **string_list**.
4. **Printing Strings:** Another loop is used to iterate over the **string_list**, printing each stored string.

Functions/Data Types Used - Explanation with Example:

1. **input()** Function: This function is used to obtain user input. Example: **num_strings = int(input("Enter the number of strings: "))**.
2. Lists: Lists are data structures in Python used to store multiple items. In this program, the **string_list** is a list that stores the entered strings.

Example Syntax and Description:

```
string_list = []
num_strings = int(input("Enter the number of strings: "))
for i in range(num_strings):
    string = input(f"Enter string {i + 1}: ")
    string_list.append(string)
```

Program

```
# Create an empty list
string_list = []

# Prompt the user for input and store in the list
num_strings = int(input("Enter the number of strings: "))
for i in range(num_strings):
    string = input(f"Enter string {i + 1}: ")
    string_list.append(string)

# Print the strings stored in the list
print("Strings in the list:")
for string in string_list:
    print(string)
```

4.3) Explanation of Execution:

1. An empty list, string_list, is created.
2. The user inputs the number of strings to store.
3. The program iterates through the specified range, asking users for input. Each string is appended to string_list.

5.0) Conclusion:

This programming experiment enhances students' comprehension of data structures and input-output operations in Python. By practicing the process of gathering user input, storing strings in a list, and displaying the stored data, students acquire foundational programming skills. The outcomes of this exercise include an understanding of lists, string manipulation, and the importance of user interaction in programming.