	SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGG.	**LABORATORY MANUAL**
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE : Write a program to demonstrate working with tuples in python	

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 a program to demonstrate working with tuples in python

2.0) SCOPE: This program aims to provide a comprehensive understanding of working with tuples in Python. It covers tuple creation, element access, length determination, looping, element existence check, and tuple unpacking.

3.0) FACILITIES/ APPARATUS:

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

4.0) THEORY:

Tuples in Python:

Tuples are ordered, immutable collections of elements in Python. They are similar to lists but have the key difference that once you create a tuple, you cannot change its contents - they are immutable.

Example:

	SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGG.	**LABORATORY MANUAL**
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE : Write a program to demonstrate working with tuples in python	
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: 2 OF 2

```python
# Creating a tuple
fruits = ("apple", "banana", "cherry")

# Accessing elements
print(fruits[0])  # Output: "apple"

# Slicing a tuple
print(fruits[1:3])  # Output: ("banana", "cherry")

# Length of a tuple
print(len(fruits))  # Output: 3

# Looping through a tuple
for fruit in fruits:
    print(fruit)

# Checking if an element exists in a tuple
if "apple" in fruits:
    print("Yes, 'apple' is in the fruits tuple")

# Tuple unpacking
a, b, c = fruits
print(a, b, c)  # Output: "apple banana cherry"
```

Explanation:

1. We create a tuple named fruits containing three elements: "apple," "banana," and "cherry."
2. We access elements using indexing, just like in lists.

3. Slicing works with tuples, allowing you to extract a portion of the tuple.
4. We use the len() function to find the length of the tuple, which is 3.
5. A for loop iterates through the elements in the tuple, allowing us to print each fruit.
6. We check if "apple" exists in the tuple using the in operator.
7. Tuple unpacking assigns each element of the tuple to separate variables (a, b, and c).

Tuples are commonly used when you want to create a collection of items that should not be modified after creation, such as coordinates (x, y) or when you want to return multiple values from a function.

A tuple is an ordered, immutable collection of elements in Python. This program demonstrates various tuple operations:

1. Tuple Creation: We create a tuple named fruits containing three elements: "apple," "banana," and "cherry."
2. Element Access: Elements in a tuple are accessed using indexing. For example, fruits[0] accesses the first element, which is "apple."
3. Length Determination: The len() function is used to find the length of a tuple, which is the number of elements it contains. In this case, len(fruits) returns 3.
4. Looping Through a Tuple: A for loop is employed to iterate through the elements of the tuple and print each fruit.
5. Element Existence Check: We use the in operator to check if an element exists in the tuple. In this program, we check if "apple" exists in fruits.
6. Tuple Unpacking: Tuple unpacking assigns each element of the tuple to separate variables (a, b, and c). In this case, a, b, and c will hold "apple," "banana," and "cherry," respectively.

Example: Suppose we have a tuple named coordinates representing a point in 2D space:

```
coordinates = (3, 4)
```

We can access its elements and calculate the distance from the origin using the Pythagorean theorem:

```
x, y = coordinates  # Unpack the tuple
distance = (x**2 + y**2)**0.5  # Calculate distance from (0, 0)
print("Distance from the origin:", distance)
```

In this example, we unpack the tuple coordinates, calculate the distance from the origin using the elements x and y, and print the result.

Program

	SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGG.	**LABORATORY MANUAL**
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE : Write a program to demonstrate working with tuples in python	
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: 5 OF 2

```python
# Creating a tuple
fruits = ("apple", "banana", "cherry")

# Accessing elements
print("First fruit:", fruits[0])
print("Second fruit:", fruits[1])
print("Third fruit:", fruits[2])

# Length of the tuple
print("Number of fruits in the tuple:", len(fruits))

# Looping through the tuple
print("Fruits in the tuple:")
for fruit in fruits:
    print(fruit)

# Checking if an element exists in the tuple
if "apple" in fruits:
    print("'apple' is in the tuple")

# Tuple unpacking
a, b, c = fruits
print("Unpacked fruits:", a, b, c)
```

Output

```
First fruit: apple
Second fruit: banana
Third fruit: cherry
Number of fruits in the tuple: 3
Fruits in the tuple:
apple
banana
cherry
'apple' is in the tuple
Unpacked fruits: apple banana cherry
```

In this program, we create a tuple named fruits, access its elements, find its length, loop through the elements, check if an element exists, and demonstrate tuple unpacking. This example showcases various operations that can be performed with tuples in Python.

4.2) Program Execution:

1. We create the fruits tuple with three elements.
2. The program accesses and prints the first, second, and third elements of the tuple.
3. It calculates and displays the number of fruits in the tuple.
4. A for loop iterates through the tuple's elements, printing each fruit.
5. The program checks if "apple" is present in the tuple and displays a message accordingly.
6. Finally, tuple unpacking is demonstrated by assigning elements to variables a, b, and c.

5.0) Conclusion:

This program provides hands-on experience with tuples in Python. Students learn how to create, access, and manipulate tuples, which are essential data structures in Python. Additionally, they gain insights into practical scenarios where tuples, with their immutability and ordering, are useful for data storage and manipulation.