Experiment 06 – Build Python Program to demonstrate Data Structures in Python (List, Tuples, Vectors, Data Frames).

Learning Objective: Student should be able to build a program through which they can demonstrate Different kinds of Data Structures in Python like List, Tuples & Vectors Etc.

Tools: Python under Windows environment

Theory: Develop a python program demonstrating Data Structures in Python

1. Write A Python Program showcasing different kinds of Python Data Structures:

Here are the general steps for building a Python demonstrating Data Structures in Python:

1. Once you have installed Python, open a text editor such as Notepad or Sublime Text.
2. Create a new Python file and save it with a .py extension.
3. Start by importing the necessary libraries. For example, you can import pandas to work with data frames:

```
import pandas as pd
```

4. Next, you can define a list and a tuple:

```
my_list = [1, 2, 3, 4, 5]
my_tuple = (6, 7, 8, 9, 10)
```

5. You can then use the list and tuple to perform operations such as slicing, adding, removing, and updating elements.

```
# slicing the list
print(my_list[1:4])

# adding an element to the list
my_list.append(6)

# removing an element from the list
my_list.remove(3)

# updating an element in the list
my_list[2] = 0
```

6. Next, you can define a vector using the NumPy library:

```
import numpy as np

my_vector = np.array([1, 2, 3, 4, 5])
```

7. You can then perform operations on the vector, such as adding, multiplying, and taking the dot product.

```
# adding a scalar to the vector
my_vector = my_vector + 2

# multiplying the vector by a scalar
my_vector = my_vector * 3

# taking the dot product of two vectors
my_vector2 = np.array([6, 7, 8, 9, 10])
dot_product = np.dot(my_vector, my_vector2)
```

8. Finally, you can create a data frame using the pandas library:

```
my_data = {
    'name': ['John', 'Sarah', 'Bill', 'Jane', 'Steve'],
    'age': [25, 30, 35, 40, 45],
    'gender': ['M', 'F', 'M', 'F', 'M']
}

df = pd.DataFrame(my_data)
```

10. You can then perform operations on the data frame, such as selecting specific columns or rows, filtering data based on conditions, and sorting the data.

```
# selecting specific columns
df[['name', 'age']]

# selecting specific rows based on a condition
df[df['age'] > 30]

# sorting the data by age in descending order
df.sort_values(by='age', ascending=False)
```

Program:

```
import numpy as np

# defining a list of numbers
my_list = [1, 2, 3, 4, 5]

# printing the length of the list
print("Length of the list:", len(my_list))

# adding an element to the list
my_list.append(6)

# removing an element from the list
my_list.remove(3)

# updating an element in the list
my_list[2] = 0

# printing the updated list
print("Updated list:", my_list)

# defining a tuple of names
my_tuple = ("John", "Sarah", "Bill", "Jane", "Steve")

# printing the first and last elements of the tuple
print("First name in the tuple:", my_tuple[0])
print("Last name in the tuple:", my_tuple[-1])

# defining a vector using NumPy
my_vector = np.array([1, 2, 3, 4, 5])

# adding a scalar to the vector
my_vector = my_vector + 2

# multiplying the vector by a scalar
my_vector = my_vector * 3

# printing the final vector
print("Final vector:", my_vector)
```

Output:

```
Length of the list: 5
Updated list: [1, 2, 0, 5, 6]
First name in the tuple: John
Last name in the tuple: Steve
Final vector: [ 9 12 15 18 21]
```

Learning Outcomes: The student should have the ability to implement Data Structures in Python

LO1: Learnt How to Explore different kinds of Data Structures in Python.

LO2: In this experiment, we have imported modules & then created a List, Tuple & made use of Vectors & get the desired output.

Course Outcomes: Upon completion of the course students will be execute programs in Python & Applied different kinds of Data structures in Python.

Conclusion: In this Experiment, we have learnt to implement:

1. **Various Kinds of Data Structures like List, Tuples & Vectors etc.**

2. **We have learnt in this experiment how to import modules & made use of them to create List, Tuples & Vectors. Using these modules, we have performed different kinds of operations.**

For Faculty Use

Correction Parameters	**Formative Assessment [40%]**	**Timely completion of Practical [40%]**	**Attendance / Learning Attitude [20%]**	
Marks Obtained				