# S. Y. B. Tech (ECE)

**Trimester: VI Subject: Linux Based Python Laboratory (CET2005A)**

# Name: Class:

**Roll No.: Batch:**

# Experiment – 05

**Title: Introduction to Advanced Data Structures of Python (Any two)**

# Performed on:

**Marks**

**Teacher’s Signature with date**

**Submitted on:**

**Aim**: Introduction to Advanced Data Structures of Python (Any two)

# Objective:

* 1. To know the Advanced Data Structures of Python.
  2. To implement at least two Advanced Data Structures of Python programs.

# Theory:

* **Advanced Data Structures in Python:**

# Tuple

* **Functions**

# List and set

* **Sorting**

# Dictionary Tuple in python

**A tuple is another sequence data type that is similar to the list. The main differences between lists and tuples are:**

# Lists are enclosed in brackets ( [ ] ) and their elements and size can be changed, while tuples are enclosed in parentheses ( ( ) ) and cannot be updated.

**Tuple example**

# TupSub = ( ‘ADC', ‘MC’ , ‘EE322’ )

**TupMob = (‘Iphone6’,Sony‘,’Appo’)**

# Print (TupSub) # Prints complete list

**Print (TupSub[0]) # Prints first element of the list**

# print (TupSub[1:3]) # Prints elements starting from 2nd till 3rd Print (TupSub[2:]) # Prints elements starting from 3rd element Print (TupMob \* 2 ) # Prints list two times

**print (TupSub + TupMob) # Prints concatenated lists Example : Program to merge two unsorted lists**

# A = [100,50,150]

**B = [9,51,20,3]**

# Merged sorted list C=[3,9,20,50,51,100,150] Step 1: Create two user input list.

**Step 2: Final merge list size is (size of the first list + the size of the second list). Step 3: Sort two lists using sort() method.**

# Step 4: Merge two sorted list and store it into a third list.

**Step 5: Merging remaining elements of a[] (if any).Merging remaining elements of b[] (if any).**

# Step 6: Display merged sorted list.

**Step 1: A = [100,50,150] B = [9,51,20,3]**

# Step 2: Final merged list is C and its size is (3+4). Step 3: Sort two lists using sort() method.

**A = [50,100,150] B = [3,9,20,51]**

# Step 4: Merge two sorted list and store it into a third list.

**C=[3,9,20,50,51]**

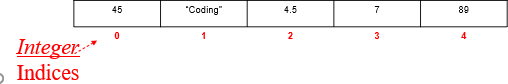
# Step 5: Merging remaining elements of A or B (if any).

**C=[3,9,20,50,51,100,150]**

# Step 6: Display merged sorted list.

**Dictionaries**

# Lists, tuples, and strings hold elements with only integer indices



**In essence, each element has an index (or a key) which can only be an integer**

# What if we want to store elements with non-integer indices (or keys)?

**Python Dictionary**

# Python 's dictionaries are hash table type. They work like associative arrays or hashes found in Perl and consist of key-value pairs.

**Keys can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object.**

# Dictionaries are enclosed by curly braces ( { } ) and values can be assigned and accessed using square braces ( [] ).

**Dictionaries**

# Lists index their entries based on the position in the list Dictionaries are like bags - no order

**So we index the things we put in the dictionary with a “lookup tag”**

# >>> purse = dict()

**>>> purse['money'] = 12**

# >>> purse['candy'] = 3

**>>> purse['tissues'] = 75**

# >>> print purse

**{'money': 12, 'tissues': 75, 'candy': 3}**

# >>> print purse['candy'] 3

**>>> purse['candy'] = purse['candy'] + 2’**

# >>> print purse

**{'money': 12, 'tissues': 75, 'candy': 5}**

# Example of dictionary

**dict = {}**

# dict['one'] = "This is one" dict[2] = "This is two“

**tinydict = {'name': 'john','code':6734, 'dept': 'sales'} print dict['one'] # Prints value for 'one' key print dict[2] # Prints value for 2 key**

# print tinydict # Prints complete dictionary print tinydict.keys() # Prints all the keys

**print tinydict.values() # Prints all the values OUTPUT:**

# This is one This is two

**{'dept': 'sales', 'code': 6734, 'name': 'john'} ['dept', 'code', 'name']**

# ['sales', 6734, 'john']

**Input:** Pyhton Programs. **Output:** Output of each Program. **Conclusion:**

# Post Lab Questions:

1. **What are advanced data structures in python? Explain in detail.**

# Explain in detail Queues in Python

1. **Write a program in python to Print the middle element of a given linked list**

# Remove duplicate elements from a sorted linked list

**Additional Reference Links:**

1. [https://www.python.org](https://www.python.org/)
2. [https://www.tutorialspoint.com](https://www.tutorialspoint.com/)
3. <https://www.programiz.com/python-programming>