**Deccan Education Society's (DES)**
**Pune University, Pune**
**School of Engineering and Technology**
**Department of Computer Engineering and Technology**
**Program: B. Tech in Computer Science and Engineering**

| | | |
|---|---|---|
| **Academic Year:** 2024-25 | **Year:** Third Year | **Term:** II |
| **PRN No.: 1012412079** | **Name: Ratnajeet Patil** | |
| **Subject:** Python Programming | | |
| **Assignment No.**: 1 | | |
| **Date:** | | |

# Lab Assignment: Practice

## 1. Finding the Length of a Tuple

```
my_tuple = (1, 2, 3, 4, 5)
print(f"Tuple: {my_tuple}")
print(f"Length: {len(my_tuple)}")
```

✓ 0.0s

```
Tuple: (1, 2, 3, 4, 5)
Length: 5
```

## 2. Converting a List to a Tuple and Vice Versa

```
# List to Tuple
my_list = [1, 2, 3, 4, 5]
my_tuple = tuple(my_list)
print(f"List to Tuple: {my_list} → {my_tuple}")

# Tuple to List
my_tuple = ('a', 'b', 'c', 'd')
my_list = list(my_tuple)
print(f"Tuple to List: {my_tuple} → {my_list}")
```

✓ 0.0s

```
List to Tuple: [1, 2, 3, 4, 5] → (1, 2, 3, 4, 5)
Tuple to List: ('a', 'b', 'c', 'd') → ['a', 'b', 'c', 'd']
```

## 3. Finding Maximum and Minimum Values in a Tuple

```
numbers = (23, 45, 12, 67, 8, 34)
print(f"Tuple: {numbers}")
print(f"Maximum value: {max(numbers)}")
print(f"Minimum value: {min(numbers)}")
```

[20] ✓ 0.0s

```
Tuple: (23, 45, 12, 67, 8, 34)
Maximum value: 67
Minimum value: 8
```

## 4. Concatenating Two Tuples

```
tuple1 = (1, 2, 3)
tuple2 = (4, 5, 6)
result = tuple1 + tuple2
print(f"{tuple1} + {tuple2} = {result}")
```

[21] ✓ 0.0s

```
(1, 2, 3) + (4, 5, 6) = (1, 2, 3, 4, 5, 6)
```

## 5. Checking if an Element Exists Within a Tuple

```
my_tuple = ('apple', 'banana', 'cherry', 'orange')
element1 = 'banana'
element2 = 'grape'

print(f"Is '{element1}' in {my_tuple}? {element1 in my_tuple}")
print(f"Is '{element2}' in {my_tuple}? {element2 in my_tuple}")
```

[22] ✓ 0.0s

```
Is 'banana' in ('apple', 'banana', 'cherry', 'orange')? True
Is 'grape' in ('apple', 'banana', 'cherry', 'orange')? False
```

## 6. Finding the Index of a Specific Element in a Tuple

```
fruits = ('apple', 'banana', 'cherry', 'orange', 'banana')
element = 'banana'
index = fruits.index(element)
print(f"The first occurrence of '{element}' is at index {index}")

# Using try-except for element not in tuple
try:
    print(f"Index of 'grape': {fruits.index('grape')}")
except ValueError:
    print("'grape' is not in the tuple")
```

23] ✓ 0.0s

```
The first occurrence of 'banana' is at index 1
'grape' is not in the tuple
```

## 7. Unpacking the Elements of a Tuple into Individual Variables

```
person = ('John', 25, 'Developer', 'New York')
name, age, profession, city = person

print(f"Name: {name}")
print(f"Age: {age}")
print(f"Profession: {profession}")
print(f"City: {city}")
```

24] ✓ 0.0s

```
Name: John
Age: 25
Profession: Developer
City: New York
```

## 8. Reversing a Tuple

```
original = (1, 2, 3, 4, 5)
reversed_tuple = original[::-1]
print(f"Original: {original}")
print(f"Reversed: {reversed_tuple}")
```

5] ✓ 0.0s

```
Original: (1, 2, 3, 4, 5)
Reversed: (5, 4, 3, 2, 1)
```

## 9. Converting a Tuple of Strings into a Single Concatenated String

```
words = ('Hello', 'world', 'from', 'Python')
result = ' '.join(words)
print(f"Tuple: {words}")
print(f"Concatenated string: '{result}'")
```

6] ✓ 0.0s

```
Tuple: ('Hello', 'world', 'from', 'Python')
Concatenated string: 'Hello world from Python'
```

## 10. Sorting a Tuple of Numbers in Ascending or Descending Order

```
numbers = (5, 2, 8, 1, 9, 3)
ascending = tuple(sorted(numbers))
descending = tuple(sorted(numbers, reverse=True))

print(f"Original: {numbers}")
print(f"Ascending: {ascending}")
print(f"Descending: {descending}")
```

7] ✓ 0.0s

```
Original: (5, 2, 8, 1, 9, 3)
Ascending: (1, 2, 3, 5, 8, 9)
Descending: (9, 8, 5, 3, 2, 1)
```

## 11. Converting a List of Tuples into a Dictionary

```
tuple_list = [('a', 1), ('b', 2), ('c', 3), ('d', 4)]
result_dict = dict(tuple_list)

print(f"List of tuples: {tuple_list}")
print(f"Dictionary: {result_dict}")
```

8] ✓ 0.0s

```
List of tuples: [('a', 1), ('b', 2), ('c', 3), ('d', 4)]
Dictionary: {'a': 1, 'b': 2, 'c': 3, 'd': 4}
```

## 12. Slicing a Tuple to Get a Sub-tuple

```
my_tuple = (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
sub_tuple = my_tuple[2:7]

print(f"Original tuple: {my_tuple}")
print(f"Sub-tuple (index 2 to 6): {sub_tuple}")
```

9] ✓ 0.0s

```
Original tuple: (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
Sub-tuple (index 2 to 6): (2, 3, 4, 5, 6)
```

# List Operations

## 13. Finding the Length of a List

```
my_list = [10, 20, 30, 40, 50]
print(f"List: {my_list}")
print(f"Length: {len(my_list)}")
```

0] ✓ 0.0s

```
List: [10, 20, 30, 40, 50]
Length: 5
```

## 14. Appending an Element to the End of a List

```
fruits = ['apple', 'banana', 'cherry']
print(f"Before append: {fruits}")

fruits.append('orange')
print(f"After append: {fruits}")
```

31] ✓ 0.0s

```
Before append: ['apple', 'banana', 'cherry']
After append: ['apple', 'banana', 'cherry', 'orange']
```

## 15. Inserting an Element at a Specific Index in a List

```
numbers = [1, 2, 4, 5]
print(f"Before insert: {numbers}")

numbers.insert(2, 3)  # Insert 3 at index 2
print(f"After insert: {numbers}")
```

32] ✓ 0.0s

```
Before insert: [1, 2, 4, 5]
After insert: [1, 2, 3, 4, 5]
```

## 16. Removing the First Occurrence of a Specified Element from a List

```
fruits = ['apple', 'banana', 'cherry', 'banana', 'orange']
print(f"Before removal: {fruits}")

fruits.remove('banana')  # Remove first occurrence of 'banana'
print(f"After removal: {fruits}")

# Using try-except for element not in list
try:
    fruits.remove('grape')
except ValueError:
    print("'grape' is not in the list")
```

✓ 0.0s

```
Before removal: ['apple', 'banana', 'cherry', 'banana', 'orange']
After removal: ['apple', 'cherry', 'banana', 'orange']
'grape' is not in the list
```

## 17. Finding the Largest and Smallest Elements in a List

```
numbers = [23, 54, 12, 87, 5, 32, 68]
print(f"List: {numbers}")
print(f"Largest element: {max(numbers)}")
print(f"Smallest element: {min(numbers)}")
```

✓ 0.0s

```
List: [23, 54, 12, 87, 5, 32, 68]
Largest element: 87
Smallest element: 5
```