## 

## 

**Experiment 2:**

**Strings, Lists, Tuples, and Dictionaries**

CPE106L (Software Design Laboratory)

**Member 1: Claros, Angelica A.**

**Member 2: Facal, Ma. Catherina N.**

**Member 3: Santos, Angelica Anne D.**

Group No.: **5**

Section: **B2**

## **PreLab**



|  |
| --- |
| **Laboratory Insights** |

**Claros**

In the second laboratory, the focus was on learning about strings, lists, tuples, and dictionaries. By using the METIS book entitled "Fundamentals of Python" and documentation related to Python, such as the Python Docs webpage, I was able to grasp the essential knowledge needed to understand the basics of Python.

Initially, through the book, the process of declaring a string was illustrated. Strings hold data; they can contain alphabets, words, or even characters. I discovered that in Python, a single character is still considered a string, which is different from C++ where one letter is considered a char. Additionally, it was discovered in the lesson that in Python, you can use either double or single quotation marks to denote a string. Next, a list was introduced as a way to store data that can be changed and updated along the way in the code, similar to dictionaries, but unlike tuples. Once a tuple is created, all its data or content cannot be updated or changed. On the other hand, dictionaries are mutable and can store data in pairs; for instance, one can set 'name' as the key and 'Angelica' as the value: 'name': 'Angelica'. Moreover, additional syntax was understood, such as the use of the type() syntax to determine the data type of a specific data and len() to determine the length of data, lists, or strings.

Overall, through this laboratory, I learned the basics and characteristics of these data structures, which are essential and useful for further coding exercises. With this knowledge, it will be easier to comprehend and come up with solutions for the coding exercises in this laboratory.

**Facal**

For Lab 2, we focused on various Python data structures such as strings, lists, tuples, and dictionaries. Strings are collections of characters which can be letters, numbers, and other special characters. Lists are mutable data structures unlike tuples which are immutable data structures, meaning lists can be modified while tuples cannot. Dictionaries are an unordered collection of key-value pairs with each unique value assigned to it. We also worked with text files for the laboratory coding program. Text files are simply files that can store plain text data which can be imported into Python programs.

An important thing that I learned while studying the book, working with strings and text files in Python involves understanding encoding and decoding processes. These processes encompass various formats, emphasizing the importance of aligning the file's encoding format with the coding and compilation of the code to prevent errors. Another significant learning pertains to working with lists. While lists can accommodate diverse data types, generating new lists from existing ones necessitates employing a concise syntax, particularly when handling extensive datasets and intricate operations. Furthermore, dictionaries are mutable through loops; however, they prohibit the duplication of keys.

**Santos**

Prelab Number 2 primarily focus on exploring additional data structures within Python, namely Strings, Tuples, Lists, and Dictionaries. Through resources such as the "Fundamentals of Python" book from METIS and Python documentation, I gained a solid understanding of the foundational concepts essential for mastering Python basics.

As per the book's explanation, in Python, a string functions as a versatile data structure capable of encompassing various data types, including alphanumeric characters, letters, and numerals, and can be defined using either single or double quotes. It's noteworthy that strings are immutable, meaning their internal elements cannot be modified once created. Similarly, tuples share this immutability characteristic with strings, setting them apart from mutable data structures such as lists and dictionaries. Additionally, I encountered essential syntax elements like type() for determining data types and len() for calculating the length of data structures like lists and strings. Moreover, our exploration extended to operations concerning text files, encompassing tasks such as writing and reading textual and numerical data from files. In Python, lists offer flexibility in data storage and support operations like indexing and slicing for efficient data manipulation. Meanwhile, dictionaries represent collections of key-value pairs, facilitating swift data addition and retrieval based on keys.

This laboratory experience provided me the understanding of the basics and characteristics of various data structures, which are fundamental for further coding exercises. Thus, I feel more equipped to comprehend solutions for the coding challenges presented in our laboratory.

**Answers to Questions**

1. b. 20

2. b. [20, 30]

3. a. 1

4. b. [10, 20, 30, 40, 50]

5. b. [10, 5, 30]

6. c. [10, 15, 20, 30]

7. b. ["name", "age"]

8. b. None

9. b. pop

10. b.