## 

## 

**Experiment 2:**

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

CPE106L (Software Design Laboratory)

Group No.: **10**

Section: **B2**

## **Prelab**



|  |
| --- |
| **Readings, Insights, and Reflection**  Note: All text in red should be removed in the final lab report. Change the font color from RED to black.  < **What to Include?>**  Readings include METIS books, pertinent websites. Provide below your Insights and Reflection. Paragraph format> |
| ***<Include ISBNs and pages of used METIS books and pertinent URLs>***  **Insights and Reflections**  ***<CANDA (CHAPTER 4 AND 5)>***   * The fundamental structure of a string and techniques for extracting substrings are covered in further depth in Chapter 4 of Fundamentals of Python: First Programs and Data Structures. The author walks you through the process of opening a file, reading data from it, and closing it. The chapter also covers using statements for file input and output and try/except blocks to handle file exceptions. A data structure is a composite entity composed of several other data objects; in contrast to an integer, which can be broken down into smaller components, a string is a form of data structure. A collection of one-to-many characters is called a string. The string data structure is unchangeable. It is consequently possible to access the characters that comprise its internal data components, but they cannot be altered, expanded, or removed. * The author of Foundations of Python: First Programs and Data Structures, Chapter 5, explains how to utilize the dictionary and the list, two more popular data structures. A list allows the programmer to control a series of data values of any kind. A dictionary organizes data values based on relationships with other data values rather than sequential order. Lists and dictionaries are useful tools for efficiently organizing data in useful and interesting applications. This chapter gives an overview of basic functions and looks at dictionaries and lists. These functions support the arrangement of computer instructions like data structures. * Generally, chapters 4 and 5 offer a fundamental framework for creating increasingly sophisticated Python programs. These chapters offer fundamental ideas that cut across many different fields of study. Developers may write more successful and productive code and be better equipped to tackle challenging programming tasks by grasping and using these ideas.   <**COLLAMAT (CHAPTER 4-5)>**   * Chapter 4 delved deeper into strings and text files compared to previous chapters. Strings were explored as sequences of characters and immutable data structures. The chapter covered determining string length using len(), accessing characters via the subscript operator [], enabling modifications and slicing. It introduced methods as operations used with objects. Text files were defined as software objects for data transfer, storage, or exportation, employing methods like write(), read(), and readline() for file manipulation. Chapter 5 shifted focus to lists and dictionaries. Lists were defined as mutable sequences of elements of any type, highlighting functions like append() and sort() for efficient list manipulation - useful for handling large data collections. The discussion then transitioned to dictionaries, unordered collections of key-value pairs, explaining methods like keys(), values(), items(), along with construction, modification, and iteration processes. Overall, these chapters provided crucial Python fundamentals insights, particularly helpful for programming tasks involving arrays, lists, or accessing files from permanent storage sources.   <**ESTACION (CHAPTER 4-5)>**   * Chapters 4 and 5 of "Fundamentals of Python: The first programs in Python are called "First Programs," they introduce you to the fundamental concepts of working with strings, text files, lists, and dictionaries. Chapter 4 illustrates string creation and manipulation by utilizing methods like lower(), upper(), and strip(), as well as string interpolation with the format() method and file operations such as opening, reading, and closing files by employing with statements and try/except blocks for resource management and error handling. Chapter 5 explains how to create and modify lists and dictionaries using methods like append(), sort(), keys(), values(), and items() and describes the efficiency of nesting lists and dictionaries for simple code. These chapters lay the foundation for learners regarding the skills necessary for more advanced programming in different fields such as data science, machine learning, and software development. |

<**TRINIDAD (CHAPTER 4-5)>**

* Chapters 4 and 5 elaborate on the fundamental ideas of Python by delving further into the fields of data structure and text manipulation. The primary form of textual data, strings are the subject of Chapter 4. We discover that strings are persistent data structures rather than just collections of characters. This chapter gives us the ability to utilize slicing, an effective technique for obtaining subsets of a string, to edit sections and even access individual characters in addition to determining the length of the string. Furthermore, the concept of methods—functions linked to objects—is presented. This opens the door to investigating built-in string methods that carry out typical text manipulation tasks, including case conversion or substring discovery.
* The fifth chapter of the book discusses dictionaries and lists, two essential Python data structures. The first part of the chapter introduces the idea of lists, which are collections of ordered things, and explains how to create, modify, and iterate over them. It also covers list operations such as sort.() and append(). This is followed by an explanation of dictionaries, which are unordered sets of key-value pairs. The author explains how to construct, maintain, modify, and iterate throughout dictionaries in addition to using dictionary functions like keys(), values(), and items. This chapter also covers nesting lists and dictionaries inside of one another, as well as the use of list and dictionary methods for simple programming.
* We can gain powerful tools for solving practical programming difficulties by understanding these essential Python fundamentals, especially the manipulation of strings, files, lists, and dictionaries. Programs with the ability to manage big datasets, interpret text, and work with data kept in fixed locations like files are built using these data structures as part of their foundation.

**Answers to Questions**

1. B
2. B
3. A
4. B
5. B
6. C
7. B
8. B
9. B
10. B