| Class Name | TE CSE-B | |
|---|---|---|
| Student Name | Yash Dhanawade | |
| Grade and Subject Teacher's Signature | | |

| Experiment Number | 12 | |
|---|---|---|
| Experiment Title | List and Dictionaries. | |
| Resource | Hardware: | Software: |
| Objective | | |

| Theory and Code | css |
|---|---|
| | Python List: A list is a mutable and ordered collection of elements enclosed in square brackets []. Lists are one of the most versatile data types in Python, as they can hold different data types such as strings, integers, floats, and other lists. Here is an example of a Python list: |
| | my_list = ["apple", 5, 3.14, ["nested", "list"]] |
| | We can access the elements of a list using indexing. Indexing starts from 0, which means the first element of a list is at index 0, the second element is at index 1, and so on. We can also access elements from the end of the list using negative indexing, where -1 refers to the last element, -2 refers to the second last element, and so on. Here are some examples of indexing: |
| | my_list[0]  # returns "apple |
| | " my_list[1]  # returns 5 |
| | my_list[-1] # returns ["nested", "list"] |
| | Lists are mutable, which means we can modify their contents by assigning new values to specific elements or using built-in methods like append(), insert(), remove(), and pop(). Here are some examples: |
| | Python Dictionary: A dictionary is a mutable and unordered collection of key-value pairs enclosed in curly braces {}. Dictionaries are used to store and retrieve data based on a unique key instead of an index, making |

| | them ideal for fast lookups. Here is an example of a Python dictionary: |
|---|---|
| | |
| Conclusion | In conclusion, lists and dictionaries are two fundamental data structures in Python that are used for storing and manipulating collections of data. Lists are mutable and ordered collections of elements, while dictionaries are mutable and unordered collections of key-value pairs. Both data structures offer a variety of built-in methods for modifying and accessing their contents, and they can be combined and nested to create more complex data structures. Understanding how to use lists and dictionaries is essential for any Python programmer. |