**Juan, Maria Evita M.**

**BSCS-3B**

1. **What is dictionary? Give example**

In Python, a dictionary is a built-in data type that stores collections of keys and values. Each key in a dictionary is unique, and it is used to access the data it represents. Dictionaries are changeable, which means that their contents can be modified after construction.

A screenshot of a computer program

Description automatically generatedExample:

1. Write a python script to traverse a dictionary

A screenshot of a computer

Description automatically generated

1. Explain 5 list built in methods of dictionary

* dict.get(key, default=None): This method returns the value for the specified key if it exists in the dictionary. If the key is not found, it returns the default value provided (which defaults to None if not specified). It's useful when you want to retrieve a value but don't want to raise a KeyError if the key is not present.
* dict.keys(): This method returns a view object that displays a list of all the keys in the dictionary. It allows you to access the keys of the dictionary and iterate over them.
* dict.values(): This method returns a view object that displays a list of all the values in the dictionary. Similar to dict.keys(), it allows you to access the values of the dictionary and iterate over them.
* dict.items(): This method returns a view object that displays a list of tuples, where each tuple contains a key-value pair from the dictionary. It allows you to access both keys and values simultaneously and iterate over them.
* dict.update(other\_dict): This method updates the dictionary with the key-value pairs from another dictionary or an iterable of key-value pairs. If a key exists in both dictionaries, the value from the other dictionary replaces the existing value in the original dictionary. If the key doesn't exist, it's added to the dictionary.

1. What is key value pair? Explain

Key-value pairs help you organize and access data more efficiently. By specifying a key, you can rapidly access the associated value without knowing where it is in the data structure. This makes dictionaries (or comparable structures in other computer languages) adaptable and valuable in a variety of applications.