Chapter 11 - 6Dictionary

Theory:

Dictionaries store key-value pairs and are unordered collections.

Code Example:

```
# ■ Python Dictionary Basics for Beginners
# ----------------------------------------
# A dictionary is used to store data in key-value pairs in Python.
# ■ DIFFERENCE: Dictionary vs List/Tuple/String
# - Dictionary: Unordered, mutable, indexed by unique keys.
# - List/Tuple/String: Ordered, indexed by position (0, 1, 2...), store single values.
#Empty Dic
d = {}
# ■ Example: Create a dictionary
marks = {
"milan": 100,
"rohan": 10,
"shubham": 23,
}
print(marks) # Prints full dictionary
print(type(marks)) #
print(marks["milan"]) # Access value of key 'milan'
# ■ Dictionary Properties:
# 1. Unordered (from Python 3.7+ insertion order is preserved)
# 2. Mutable (we can add/update/delete items)
# 3. Indexed using unique keys
# 4. Cannot contain duplicate keys (last value will overwrite)
# 5. Access type -> via key (e.g., marks["milan"]) / but in list -tuple - string access via
"index"
# ----------------------------------------
# ■ COMMON DICTIONARY METHODS
# ----------------------------------------
# ■ Access value by key using get() [safe]
print(marks.get("rohan")) # Output: 10
print(marks.get("raj")) # Output: None (no error if key not present)
# ■ Access using [] (will raise error if key not found)
# print(marks["raj"]) # ■ KeyError if 'raj' not in dictionary
# ■ Add or Update values
marks["raj"] = 50 # Add new key 'raj'
marks["milan"] = 95 # Update value for 'milan'
# ■ Remove specific item
marks.pop("shubham") # Removes key 'shubham'
# marks.pop("unknown") # ■ KeyError if key not present
# ■ Remove last inserted item
marks.popitem()
# ■ Check if a key exists
if "rohan" in marks:
print("Rohan is present")
# ■ Get all keys
print(marks.keys()) # dict_keys(['milan', 'rohan', 'raj'])
# ■ Get all values
print(marks.values()) # dict_values([95, 10, 50])
# ■ Get all key-value pairs
```

```
print(marks.items()) # dict_items([('milan', 95), ...])
# ■ Copy the dictionary
new_marks = marks.copy()
# ■ Clear all items from dictionary
marks.clear()
# ■ Create new dictionary from list of keys
subjects = ["math", "science", "english"]
default_marks = dict.fromkeys(subjects, 0)
print(default_marks) # {'math': 0, 'science': 0, 'english': 0}
# ----------------------------------------
# ■ More Dictionary Examples
# ----------------------------------------
# Dictionary with a number key and string value
marks = {
"milan": 100,
"rohan": 10,
"shubham": 23,
0: "harry", # Key can be int or string
}
print(marks)
print(marks.items()) # All key-value pairs
print(marks.keys()) # All keys
print(marks.values()) # All values
# ■ Update dictionary using update() method
marks.update({"milan": 99}) # Update existing key
marks.update({"renuka": 120}) # Add new key
print(marks)
# ■ Accessing values safely
print(marks.get("milan")) # ■ Returns value 99
print(marks.get("xyz")) # ■ Returns None if key not found
# ■■ Direct access using [] will cause error if key is missing
# print(marks["xyz"]) # ■ KeyError: 'xyz'
```