

*sub\_marks* as the key and combine it with *values()* method using dot notation ⑤. Even though a single level of nesting dictionaries may prove useful, having multiple levels of nesting may make the code unreadable.

## 7.5 The *del* Statement

To delete the *key:value* pair, use the *del* statement followed by the name of the dictionary along with the *key* you want to delete.

*del dict\_name[key]*

```
1. >>> animals = {"r": "raccoon", "c": "cougar", "m": "moose"}  
2. >>> animals  
{'r': 'raccoon', 'c': 'cougar', 'm': 'moose'}  
3. >>> del animals["c"]  
4. >>> animals  
{'r': 'raccoon', 'm': 'moose'}
```

In the *animals* ①–② dictionary, you can remove the *key:value* pair of "c": "cougar" as shown in ③.

## 7.6 Summary

- A dictionary associates a set of keys with values.
- The built-in function *dict()* returns a new dictionary initialized from an optional keyword argument and a possibly empty set of keyword arguments.
- The *for* loop is used to traverse all the keys in the dictionary.
- The *del dictionaryName[key]* statement is used to delete an item for the given key.
- Dictionary methods like *keys()*, *values()*, and *items()* are used to retrieve the values.
- Methods like *pop()* and *update()* are used to manipulate the dictionary *key:value* pairs.

## Multiple Choice Questions

1. Which of the following statements create a dictionary?
  - a. *dic = {}*
  - b. *dic = {"charles":40, "peterson":45}*
  - c. *dic = {40: "charles", 45: "peterson"}*
  - d. All of the above

2. Read the code shown below carefully and pick out the keys.

```
dic = {"game":40, "thrones":45}
```

- a. "game", 40, 45, and "thrones"
- b. "game" and "thrones"
- c. 40 and 45
- d. dic = (40: "game", 45: "thrones")

3. Gauge the output of the following code snippet.

```
fruit = {"apple":"red", "guava":"green"}
```

```
"apple" in fruit
```

- a. True
- b. False
- c. None
- d. Error

4. Consider phone\_book = {"Kalpana":7766554433, "Steffi":4499551100}. To delete the key "Kalpana" the code used is

- a. phone\_book.delete("Kalpana":7766554433)
- b. phone\_book.delete("Kalpana")
- c. del phone\_book["Kalpana"]
- d. del phone\_book("Kalpana":7766554433)

5. Assume d = {"Guido":"Python", "Dennis":"C"}. To obtain the number of entries in dictionary the statement used is

- a. d.size()
- b. len(d)
- c. size(d)
- d. d.len()

6. Consider stock\_prices = {"IBM":220, "FB":800}. What happens when you try to retrieve a value using the statement stock\_prices["IBM"]?

- a. Since "IBM" is not a value in the set, Python raises a KeyError exception.
- b. It executes fine and no exception is raised
- c. Since "IBM" is not a key in the set, Python raises a KeyError exception.
- d. Since "IBM" is not a key in the set, Python raises a syntax error.

7. Which of the following statement is false about the dictionary?

- a. The values of a dictionary can be accessed using keys.
- b. The keys of a dictionary can be accessed using values.
- c. Dictionaries are not ordered.
- d. Dictionaries are mutable.

g. What is the output of the following code?

```
stuff = {"book": "Java", "price": 45}  
stuff.get("book")
```

- a. 45
- b. True
- c. Java
- d. price

g. predict the output of the following code.

```
fish = {"g": "Goldfish", "s": "Shark"}  
fish.pop(s)  
print(fish)
```

- a. {'g': 'Goldfish', 's': 'Shark'}
- b. {'g': 'Shark'}
- c. {'g': 'Goldfish'}
- d. Error

h. The method that returns the value for the key present in the dictionary and if the key is not present then it inserts the key with default value into the dictionary.

- a. update()
- b. fromkeys()
- c. setdefault()
- d. get()

i. Guess the output of the following code.

```
grades = {90: "S", 80: "A"}
```

```
del grades
```

- a. Method del doesn't exist for the dictionary.
- b. del deletes the values in the dictionary.
- c. del deletes the entire dictionary.
- d. del deletes the keys in the dictionary.

j. Assume dic is a dictionary with some key:value pairs. What does dic.popitem() do?

- a. Removes an arbitrary key:value pair
- b. Removes all the key:value pairs
- c. Removes the key:value pair for the key given as an argument
- d. Invalid method

13. What will be the output of the following code snippet?

```
numbers = {}  
letters = {}  
comb = {}  
numbers[1] = 56  
numbers[3] = 7  
letters[4] = "B"  
comb["Numbers"] = numbers  
comb["Letters"] = letters  
print(comb)
```

- a. Nested dictionary cannot occur
- b. 'Numbers': {1: 56, 3: 7}
- c. {'Numbers': {1: 56}, 'Letters': {4: 'B'}}
- d. {'Numbers': {1: 56, 3: 7}, 'Letters': {4: 'B'}}

14. Gauge the output of the following code.

```
demo = {1: 'A', 2: 'B', 3: 'C'}  
del demo[1]  
demo[1] = 'D'  
del demo[2]  
print(len(demo))
```

- a. 0
- b. 2
- c. Error
- d. 1

15. Assuming *b* to be a dictionary, what does *any(b)* do?

- a. Returns True if any key of the dictionary is True.
- b. Returns False if dictionary is empty.
- c. Returns True if all keys of the dictionary are True.
- d. Method *any()* doesn't exist for dictionary.

16. Infer the output of the following code.

```
count = {}  
count[(1, 2, 4)] = 5  
count[(4, 2, 1)] = 7  
count[(1, 2)] = 6  
count[(4, 2, 1)] = 2  
tot = 0
```

```
for i in count:  
    tot = tot + count[i]  
print(len(count)+tot)
```

- a. 25
- b. 17
- c. 16
- d. Error

17. The \_\_\_\_\_ function returns Boolean True value if all the keys in the dictionary are True else returns False.

- a. all()
- b. sorted()
- c. len()
- d. any()

18. Predict the output of the following code.

```
>>> dic = {}  
>>> dic.fromkeys([1,2,3], "check")
```

- a. Syntax error
- b. {1: 'check', 2: 'check', 3: 'check'}
- c. 'check'
- d. {1:None, 2:None, 3:None}

19. For dictionary d = { "plum ":0.66, "pears ":1.25,"oranges ":0.49}, which of the following statement correctly updates the price of oranges to 0.52?

- a. d[2] = 0.52
- b. d[0.49] = 0.52
- c. d["oranges "] = 0.52
- d. d["plum "] = 0.52

20. The syntax that is used to modify or add a new key: value pair to a dictionary is:

- a. dictionary\_name[key] = value
- b. dictionary\_name[value] = key
- c. dictionary\_name(key) = value
- d. dictionary\_name{key} = value

21. Which of the following cannot be used as a key in Python dictionaries?

- a. Strings
- b. Lists
- c. Tuples
- d. Numerical values

22. Guess the output of the following code.

```
week = {1:"sunday", 2:"monday", 3:"tuesday"}  
for i,j in week.items():
```

```
    print(i,j)
```

✓a. 1 sunday 2 monday 3 Tuesday

b. 1 2 3

c. sunday monday tuesday

d. 1:"sunday" 2:"monday" 3:"tuesday"

23. Predict the output of the following code.

```
a = {1: "A", 2: "B", 3: "C"}  
b = {4: "D", 5: "E"}  
a.update(b)
```

```
print(a)
```

a. {1: 'A', 2: 'B', 3: 'C'}

b. Error

c. {4: 'D', 5: 'E'}

✓d. {1: 'A', 2: 'B', 3: 'C', 4: 'D', 5: 'E'}

---

## Review Questions

1. Define a dictionary. What are the advantages of using dictionary over lists.

2. Briefly explain how a dictionary is created with an example.

3. Write short notes on the following methods.

a. keys()

b. values()

✓c. get(key)

d. clear()

4. Explain nested dictionaries with an example.

5. Write a function that prompts the user for the average temperature for each day of the week and returns a dictionary containing the entered information.

6. Write a Python program to input information about a few employees as given below:

✓a. Name

b. Employee Id

c. Salary

The program should output the employee ID and salary of a specified employee, given his name.