Review for chapter 10

**Review Questions**

**Multiple Choice**

1. You can use the \_\_\_\_\_\_\_\_\_ operator to determine whether a key exists in a dictionary.

a. &

b. in

c. ^

d. ?

2. You use \_\_\_\_\_\_\_\_\_ to delete an element from a dictionary.

a. The remove method

b. The erase method

c. The delete method

d. The del statement

3. The \_\_\_\_\_\_\_\_\_ function returns the number of elements in a dictionary:

a. size()

b. len()

c. elements()

d. count()

4. You can use \_\_\_\_\_\_\_\_\_ to create an empty dictionary.

a. {}

b. ()

c. []

d. empty()

Review Questions **413**

5. The \_\_\_\_\_\_\_\_\_ method returns a randomly selected key-value pair from a dictionary.

a. pop()

b. random()

c. popitem()

d. rand\_pop()

6. The \_\_\_\_\_\_\_\_\_ method returns the value associated with a specified key, and removes

that key-value pair from the dictionary.

a. pop()

b. random()

c. popitem()

d. rand\_pop()

7. The \_\_\_\_\_\_\_\_\_ dictionary method returns the value associated with a specified key. If

the key is not found, it returns a default value.

a. pop()

b. key()

c. value()

d. get()

8. The \_\_\_\_\_\_\_\_\_ method returns all of a dictionary’s keys and their associated values as

a sequence of tuples.

a. keys\_values()

b. values()

c. items()

d. get()

9. The following function returns the number of elements in a set:

a. size()

b. len()

c. elements()

d. count()

10. You can add one element to a set with this method.

a. append

b. add

c. update

d. merge

11. You can add a group of elements to a set with this method.

a. append

b. add

c. update

d. merge

12. This set method removes an element but does not raise an exception if the element is

not found.

a. remove

b. discard

c. delete

d. erase

**414** Chapter 10 Dictionaries and Sets

13. This set method removes an element and raises an exception if the element is not found.

a. remove

b. discard

c. delete

d. erase

14. This operator can be used to find the union of two sets.

a. |

b. &

c. -

d. ^

15. This operator can be used to find the difference of two sets.

a. |

b. &

c. -

d. ^

16. This operator can be used to find the intersection of two sets.

a. |

b. &

c. -

d. ^

17. This operator can be used to find the symmetric difference of two sets.

a. |

b. &

c. -

d. ^

**True or False**

1. The keys in a dictionary must be mutable objects.

2. Dictionaries are not sequences.

3. A tuple can be a dictionary key.

4. A list can be a dictionary key.

5. The dictionary method popitem does not raise an exception if it is called on an empty

dictionary.

6. The following statement creates an empty dictionary:

mydct = {}

7. The following statement creates an empty set:

myset = ()

8. Sets store their elements in an unordered fashion.

9. You can store duplicate elements in a set.

10. The remove method raises an exception if the specified element is not found in the

set.

Review Questions **415**

**Short Answer**

1. What will the following code display?

dct = {'Monday':1, 'Tuesday':2, 'Wednesday':3}

print(dct['Tuesday'])

2. What will the following code display?

dct = {'Monday':1, 'Tuesday':2, 'Wednesday':3}

print(dct.get('Monday', 'Not found'))

3. What will the following code display?

dct = {'Monday':1, 'Tuesday':2, 'Wednesday':3}

print(dct.get('Friday', 'Not found'))

4. What will the following code display?

stuff = {'aaa' : 111, 'bbb' : 222, 'ccc' : 333}

print(stuff['bbb'])

5. How do you delete an element from a dictionary?

6. How do you determine the number of elements that are stored in a dictionary?

7. What will the following code display?

dct = {1:[0, 1], 2:[2, 3], 3:[4, 5]}

print(dct[3])

8. What values will the following code display? (Don’t worry about the order in which

they will be displayed.)

dct = {1:[0, 1], 2:[2, 3], 3:[4, 5]}

for k in dct:

print(k)

9. After the following statement executes, what elements will be stored in the myset set?

myset = set('Saturn')

10. After the following statement executes, what elements will be stored in the myset set?

myset = set(10)

11. After the following statement executes, what elements will be stored in the myset set?

myset = set('a bb ccc dddd')

12. After the following statement executes, what elements will be stored in the myset set?

myset = set([2, 4, 4, 6, 6, 6, 6])

13. After the following statement executes, what elements will be stored in the myset set?

myset = set(['a', 'bb', 'ccc', 'dddd'])

14. What will the following code display?

myset = set('1 2 3')

print(len(myset))

15. After the following code executes, what elements will be members of set3?

set1 = set([10, 20, 30, 40])

set2 = set([40, 50, 60])

set3 = set1.union(set2)

**416** Chapter 10 Dictionaries and Sets

16. After the following code executes, what elements will be members of set3?

set1 = set(['o', 'p', 's', 'v'])

set2 = set(['a', 'p', 'r', 's'])

set3 = set1.intersection(set2)

17. After the following code executes, what elements will be members of set3?

set1 = set(['d', 'e', 'f'])

set2 = set(['a', 'b', 'c', 'd', 'e'])

set3 = set1.difference(set2)

18. After the following code executes, what elements will be members of set3?

set1 = set(['d', 'e', 'f'])

set2 = set(['a', 'b', 'c', 'd', 'e'])

set3 = set2.difference(set1)

19. After the following code executes, what elements will be members of set3?

set1 = set([1, 2, 3])

set2 = set([2, 3, 4])

set3 = set1.symmetric\_difference(set2)

20. Look at the following code:

set1 = set([100, 200, 300, 400, 500])

set2 = set([200, 400, 500])

Which of the sets is a subset of the other?

Which of the sets is a superset of the other?

**Algorithm Workbench**

1. Write a statement that creates a dictionary containing the following key-value pairs:

'a' : 1

'b' : 2

'c' : 3

2. Write a statement that creates an empty dictionary.

3. Assume the variable dct references a dictionary. Write an if statement that determines

whether the key 'James' exists in the dictionary. If so, display the value that is associated

with that key. If the key is not in the dictionary, display a message indicating so.

4. Assume the variable dct references a dictionary. Write an if statement that determines

whether the key 'Jim' exists in the dictionary. If so, delete 'Jim' and its associated

value.

5. Write code to create a set with the following integers as members: 10, 20, 30, and 40.

6. Assume each of the variables set1 and set2 references a set. Write code that creates

another set containing all the elements of set1 and set2 and assigns the resulting set

to the variable set3.

7. Assume each of the variables set1 and set2 references a set. Write code that creates

another set containing only the elements that are found in both set1 and set2 and

assigns the resulting set to the variable set3.

Programming Exercises **417**

8. Assume each of the variables set1 and set2 references a set. Write code that creates

another set containing the elements that appear in set1 but not in set2 and assigns the

resulting set to the variable set3.

9. Assume each of the variables set1 and set2 references a set. Write code that creates

another set containing the elements that appear in set2 but not in set1 and assigns the

resulting set to the variable set3.

10. Assume each of the variables set1 and set2 references a set. Write code that creates

another set containing the elements that are not shared by set1 and set2 and assigns

the resulting set to the variable set3.

11. Assume the variable dct references a dictionary. Write code that pickles the dictionary

and saves it to a file named mydata.dat.

12. Write code that retrieves and unpickles the dictionary that you pickled in Algorithm

Workbench 11.

**Programming Exercises**

**1. Course information**

Write a program that creates a dictionary containing course numbers and the room numbers

of the rooms where the courses meet. The dictionary should have the following keyvalue

pairs:

**Course Number (key) Room Number (value)**

CS101 3004

CS102 4501

CS103 6755

NT110 1244

CM241 1411

**Course Number (key) Instructor (value)**

CS101 Haynes

CS102 Alvarado

CS103 Rich

NT110 Burke

CM241 Lee

The program should also create a dictionary containing course numbers and the names of

the instructors that teach each course. The dictionary should have the following key-value

pairs:

**418** Chapter 10 Dictionaries and Sets

The program should also create a dictionary containing course numbers and the meeting

times of each course. The dictionary should have the following key-value pairs:

**Course Number (key) Meeting Time (value)**

CS101 8:00 a.m.

CS102 9:00 a.m.

CS103 10:00 a.m.

NT110 11:00 a.m.

CM241 1:00 p.m.

The program should let the user enter a course number, and then it should display the

course’s room number, instructor, and meeting time.

**2. Capital Quiz**

Write a program that creates a dictionary containing the U.S. states as keys and their capitals

as values. (Use the Internet to get a list of the states and their capitals.) The program

should then randomly quiz the user by displaying the name of a state and asking the user

to enter that state’s capital. The program should keep a count of the number of correct and

incorrect responses. (As an alternative to the U.S. states, the program can use the names of

countries and their capitals.)

**3. File Encryption and Decryption**

Write a program that uses a dictionary to assign “codes” to each letter of the alphabet. For

example:

codes = { 'A' : '%', 'a' : '9', 'B' : '@', 'b' : '#', *etc...*}

Using this example, the letter A would be assigned the symbol %, the letter a would be

assigned the number 9, the letter B would be assigned the symbol @, and so forth.

The program should open a specified text file, read its contents, and then use the dictionary

to write an encrypted version of the file’s contents to a second file. Each character in

the second file should contain the code for the corresponding character in the first file.

Write a second program that opens an encrypted file and displays its decrypted contents on

the screen.

**4. Unique Words**

Write a program that opens a specified text file and then displays a list of all the unique

words found in the file.

*Hint: Store each word as an element of a set.*

**5. Word Frequency**

Write a program that reads the contents of a text file. The program should create a dictionary

in which the keys are the individual words found in the file and the values are the

number of times each word appears. For example, if the word “the” appears 128 times,

the dictionary would contain an element with 'the' as the key and 128 as the value.

The program should either display the frequency of each word or create a second file

containing a list of each word and its frequency.

Programming Exercises **419**

**6. File Analysis**

Write a program that reads the contents of two text files and compares them in the following

ways:

• It should display a list of all the unique words contained in both files.

• It should display a list of the words that appear in both files.

• It should display a list of the words that appear in the first file but not the second.

• It should display a list of the words that appear in the second file but not the first.

• It should display a list of the words that appear in either the first or second file but not both.

*Hint: Use set operations to perform these analyses.*

**7. World Series Winners**

In this chapter’s source code folder (available on the book’s companion Web site at www.

pearsonhighered.com/gaddis), you will find a text file named WorldSeriesWinners.txt.

This file contains a chronological list of the World Series’ winning teams from 1903

through 2009. The first line in the file is the name of the team that won in 1903, and the

last line is the name of the team that won in 2009. (Note that the World Series was not

played in 1904 or 1994. There are entries in the file indicating this.)

Write a program that reads this file and creates a dictionary in which the keys are the names

of the teams and each key’s associated value is the number of times the team has won the

World Series. The program should also create a dictionary in which the keys are the years

and each key’s associated value is the name of the team that won that year.

The program should prompt the user for a year in the range of 1903 through 2009. It

should then display the name of the team that won the World Series that year and the number

of times that team has won the World Series.

**8. Name and Email Addresses**

Write a program that keeps names and email addresses in a dictionary as key-value pairs.

The program should display a menu that lets the user look up a person’s email address, add

a new name and email address, change an existing email address, and delete an existing

name and email address. The program should pickle the dictionary and save it to a file

when the user exits the program. Each time the program starts, it should retrieve the dictionary

from the file and unpickle it.

**9. Blackjack Simulation**

Previously in this chapter you saw the card\_dealer.py program that simulates cards being

dealt from a deck. Enhance the program so it simulates a simplified version of the game of

Blackjack between two virtual players. The cards have the following values:

• Numeric cards are assigned the value they have printed on them. For example, the value

of the 2 of spades is 2, and the value of the 5 of diamonds is 5.

• Jacks, queens, and kings are valued at 10.

• Aces are valued at 1 or 11, depending on the player’s choice.

The program should deal cards to each player until one player’s hand is worth more than

21 points. When that happens, the other player is the winner. (It is possible that both player’s

hands will simultaneously exceed 21 points, in which case neither player wins.) The

program should repeat until all the cards have been dealt from the deck.

If a player is dealt an ace, the program should decide the value of the card according to the

following rule: The ace will be worth 11 points, unless that makes the player’s hand exceed

21 points. In that case, the ace will be worth 1 point.