**Review Questions**

**Multiple Choice**

1. This term refers to an individual item in a list.

a. element

b. bin

c. cubby hole

d. slot

2. This is a number that identifies an item in a list.

a. element

b. index

c. bookmark

d. identifier

3. This is the first index in a list.

a. \_1

b. 1

c. 0

d. The size of the list minus one

4. This is the last index in a list.

a. 1

b. 99

c. 0

d. The size of the list minus one

5. This will happen if you try to use an index that is out of range for a list.

a. a ValueError exception will occur

b. an IndexError exception will occur

c. The list will be erased and the program will continue to run.

d. Nothing—the invalid index will be ignored

6. This function returns the length of a list.

a. length

b. size

c. len

d. lengthof

7. When the \* operator’s left operand is a list and its right operand is an integer, the

operator becomes this.

a. The multiplication operator

b. The repetition operator

c. The initialization operator

d. Nothing—the operator does not support those types of operands.

8. This list method adds an item to the end of an existing list.

a. add

b. add\_to

c. increase

d. append

9. This removes an item at a specific index in a list.

a. The remove method

b. The delete method

c. The del statement

d. The kill method

10. Assume the following statement appears in a program:

mylist = []

Which of the following statements would you use to add the string 'Labrador' to the

list at index 0?

a. mylist[0] = 'Labrador'

b. mylist.insert(0, 'Labrador')

c. mylist.append('Labrador')

d. mylist.insert('Labrador', 0)

11. If you call the index method to locate an item in a list and the item is not found, this

happens.

a. A ValueError exception is raised

b. An InvalidIndex exception is raised

c. The method returns \_1

d. Nothing happens. The program continues running at the next statement.

12. This built-in function returns the highest value in a list.

a. highest

b. max

c. greatest

d. best\_of

13. This file object method returns a list containing the file’s contents.

a. to\_list

b. getlist

c. readline

d. readlines

14. Which of the following statements creates a tuple?

a. values = [1, 2, 3, 4]

b. values = {1, 2, 3, 4}

c. values = (1)

d. values = (1,)

**True or False**

1. Lists in Python are immutable.

2. Tuples in Python are immutable.

3. The del statement deletes an item at a specified index in a list.

4. Assume list1 references a list. After the following statement executes, list1 and

list2 will reference two identical but separate lists in memory:

list2 = list1

5. A file object’s writelines method automatically writes a newline ('\n') after writing

each list item to the file.

6. You can use the + operator to concatenate two lists.

7. A list can be an element in another list.

8. You can remove an element from a tuple by calling the tuple’s remove method.

**Short Answer**

1. Look at the following statement:

numbers = [10, 20, 30, 40, 50]

a. How many elements does the list have?

b. What is the index of the first element in the list?

c. What is the index of the last element in the list?

2. Look at the following statement:

numbers = [1, 2, 3]

a. What value is stored in numbers[2]?

b. What value is stored in numbers[0]?

c. What value is stored in numbers[-1]?

3. What will the following code display?

values = [2, 4, 6, 8, 10]

print(values[1:3])

4. What does the following code display?

numbers = [1, 2, 3, 4, 5, 6, 7]

print(numbers[5:])

5. What does the following code display?

numbers = [1, 2, 3, 4, 5, 6, 7, 8]

print(numbers[-4:])

6. What does the following code display?

values = [2] \* 5

print(values)

**338** Chapter 8 Lists and Tuples

**Algorithm Workbench**

1. Write a statement that creates a list with the following strings: 'Einstein', 'Newton',

'Copernicus', and 'Kepler'.

2. Assume names references a list. Write a for loop that displays each element of the

list.

3. Assume the lists numbers1 has 100 elements and numbers2 is an empty list. Write

code that copies the values in numbers1 to numbers2.

4. Draw a flowchart showing the general logic for totaling the values in a list.

5. Write a function that accepts a list as an argument (assume the list contains integers)

and returns the total of the values in the list.

6. Assume the names variable references a list of strings. Write code that determines

whether 'Ruby' is in the names list. If it is, display the message 'Hello Ruby'.

Otherwise, display the message 'No Ruby'.

7. What will the following code print?

list1 = [40, 50, 60]

list2 = [10, 20, 30]

list3 = list1 + list2

print(list3)

8. Write a statement that creates a two-dimensional list with 5 rows and 3 columns. Then

write nested loops that get an integer value from the user for each element in the list.

**Programming Exercises**

**1. Total Sales**

Design a program that asks the user to enter a store’s sales for each day of the week. The

amounts should be stored in a list. Use a loop to calculate the total sales for the week and

display the result.

**2. Lottery Number Generator**

Design a program that generates a seven-digit lottery number. The program should generate

seven random numbers, each in the range of 0 through 9, and assign each number to a

list element. (Random numbers were discussed in Chapter 6.) Then write another loop that

displays the contents of the list.

**3. Rainfall Statistics**

Design a program that lets the user enter the total rainfall for each of 12 months into a list.

The program should calculate and display the total rainfall for the year, the average monthly

rainfall, and the months with the highest and lowest amounts.

**4. Number Analysis Program**

Design a program that asks the user to enter a series of 20 numbers. The program should

store the numbers in a list and then display the following data:

• The lowest number in the list

• The highest number in the list

• The total of the numbers in the list

• The average of the numbers in the list

**5. Charge Account Validation**

If you have downloaded the source code from this book’s companion Web site, you will

find a file named charge\_accounts.txt in the *Chapter 08* folder. This file has a list of a

company’s valid charge account numbers. Each account number is a seven-digit number,

such as 5658845.

Write a program that reads the contents of the file into a list. The program should then ask

the user to enter a charge account number. The program should determine whether the

number is valid by searching for it in the list. If the number is in the list, the program should

display a message indicating the number is valid. If the number is not in the list, the program

should display a message indicating the number is invalid.

(You can access the book’s companion Web site at www.pearsonhighered.com/gaddis.)

**6. Driver’s License Exam**

The local driver’s license office has asked you to create an application that grades the written

portion of the driver’s license exam. The exam has 20 multiple-choice questions. Here

are the correct answers:

1. B 6. A 11. B 16. C

2. D 7. B 12. C 17. C

3. A 8. A 13. D 18. B

4. A 9. C 14. A 19. D

5. C 10. D 15. D 20. A

Your program should store these correct answers in a list. The program should read the

student’s answers for each of the 20 questions from a text file and store the answers in

another list. (Create your own text file to test the application.) After the student’s answers

have been read from the file, the program should display a message indicating whether the

student passed or failed the exam. (A student must correctly answer 15 of the 20 questions

to pass the exam.) It should then display the total number of correctly answered questions,

the total number of incorrectly answered questions, and a list showing the question numbers

of the incorrectly answered questions.

**7. Name Search**

If you have downloaded the source code from this book’s companion Web site, you will

find the following files in the *Chapter 08* folder:

• GirlNames.txt—This file contains a list of the 200 most popular names given to girls

born in the United States from the year 2000 through 2009.

• BoyNames.txt—This file contains a list of the 200 most popular names given to boys

born in the United States from the year 2000 through 2009.

Write a program that reads the contents of the two files into two separate lists. The user

should be able to enter a boy’s name, a girl’s name, or both, and the application will display

messages indicating whether the names were among the most popular.

(You can access the book’s companion Web site at www.pearsonhighered.com/gaddis.)

**8. Population Data**

If you have downloaded the source code from this book’s companion Web site, you will

find a file named USPopulation.txt in the *Chapter 08* folder. The file contains the midyear

population of the United States, in thousands, during the years 1950 through 1990. The

first line in the file contains the population for 1950, the second line contains the population

for 1951, and so forth.

Write a program that reads the file’s contents into a list. The program should display the

following data:

• The average annual change in population during the time period

• The year with the greatest increase in population during the time period

• The year with the smallest increase in population during the time period

(You can access the book’s companion Web site at www.pearsonhighered.com/gaddis.)

**9. World Series Champions**

If you have downloaded the source code from this book’s companion Web site, you will

find a file named WorldSeriesWinners.txt in the *Chapter 08* folder. 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.)

Write a program that lets the user enter the name of a team and then displays the number

of times that team has won the World Series in the time period from 1903 through 2009.

(You can access the book’s companion Web site at www.pearsonhighered.com/gaddis.)

**TIP:** Read the contents of the WorldSeriesWinners.txt file into a list. When the user

enters the name of a team, the program should step through the list, counting the number

of times the selected team appears.