Review for chapter 9

**Multiple Choice**

1. This is the first index in a string.

a. \_1

b. 1

c. 0

d. The size of the string minus one

2. This is the last index in a string.

a. 1

b. 99

c. 0

d. The size of the string minus one

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

a. a ValueError exception will occur

b. an IndexError exception will occur

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

d. Nothing—the invalid index will be ignored

4. This function returns the length of a string.

a. length

b. size

c. len

d. lengthof

5. This string method returns a copy of the string with all leading whitespace characters

removed.

a. lstrip

b. rstrip

c. remove

d. strip\_leading

**366** Chapter 9 More About Strings

6. This string method returns the lowest index in the string where a specified substring is

found.

a. first\_index\_of

b. locate

c. find

d. index\_of

7. This operator determines whether one string is contained inside another string.

a. contains

b. is\_in

c. ==

d. in

8. This string method returns true if a string contains only alphabetic characters and is at

least one character in length.

a. The isalpha method

b. The alpha method

c. The alphabetic method

d. The isletters method

9. 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.

10. This string method returns a copy of the string with all leading and trailing whitespace

characters removed.

a. clean

b. strip

c. remove\_whitespace

d. rstrip

**True or False**

1. Once a string is created, it cannot be changed.

2. You can use the for loop to iterate over the individual characters in a string.

3. The isupper method converts a string to all uppercase characters.

4. The repetition operator (\*) works with strings as well as with lists.

5. When you call a string’s split method, the method divides the string into two substrings.

**Short Answer**

1. What does the following code display?

mystr = 'yes'

mystr += 'no'

mystr += 'yes'

print(mystr)

2. What does the following code display?

mystr = 'abc' \* 3

print(mystr)

3. What will the following code display?

mystring = 'abcdefg'

print(mystring[2:5])

4. What does the following code display?

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

print(numbers[4:6])

5. What does the following code display?

name = 'joe'

print(name.lower())

print(name.upper())

print(name)

**Algorithm Workbench**

1. Assume choice references a string. The following if statement determines whether

choice is equal to ‘Y’ or ‘y’:

if choice == 'Y' or choice == 'y':

Rewrite this statement so it only makes one comparison and does not use the or operator.

(*Hint*: *use either the upper or lower methods.*)

2. Write a loop that counts the number of space characters that appear in the string referenced

by mystring.

3. Write a loop that counts the number of digits that appear in the string referenced by

mystring.

4. Write a loop that counts the number of lowercase characters that appear in the string

referenced by mystring.

5. Write a function that accepts a string as an argument and returns true if the argument

ends with the substring '.com'. Otherwise, the function should return false.

6. Write code that makes a copy of a string with all occurrences of the lowercase letter

't' converted to uppercase.

7. Write a function that accepts a string as an argument and displays the string backwards.

8. Assume mystring references a string. Write a statement that uses a slicing expression

and displays the first 3 characters in the string.

9. Assume mystring references a string. Write a statement that uses a slicing expression

and displays the last 3 characters in the string.

10. Look at the following statement:

mystring = 'cookies>milk>fudge>cake>ice cream'

Write a statement that splits this string, creating the following list:

['cookies', 'milk', 'fudge', 'cake', 'ice cream']

**368** Chapter 9 More About Strings

**Programming Exercises**

**1. Initials**

Write a program that gets a string containing a person’s first, middle, and last names, and

then display their first, middle, and last initials. For example, if the user enters John

William Smith the program should display J. W. S.

**2. Sum of Digits in a String**

Write a program that asks the user to enter a series of single-digit numbers with nothing

separating them. The program should display the sum of all the single digit numbers in the

string. For example, if the user enters 2514, the method should return 12, which is the sum

of 2, 5, 1, and 4.

**3. Date Printer**

Write a program that reads a string from the user containing a date in the form

mm/dd/yyyy. It should print the date in the form March 12, 2012.

**4. Morse Code Converter**

Morse code is a code where each letter of the English alphabet, each digit, and various

punctuation characters are represented by a series of dots and dashes. Table 9-5 shows part

of the code.

Write a program that asks the user to enter a string, and then converts that string to Morse

code.

**Table 9-5** Morse code

**Character Code Character Code Character Code Character Code**

space *space* 6 – . . . . G – – . Q – – . –

comma – – . . – – 7 – – . . . H . . . . R . – .

period . – . – . – 8 – – – . . I . . S . . .

question mark . . – – . . 9 – – – – . J . – – – T –

0 – – – – – A . – K – . – U . . –

1 . – – – – B – . . . L . – . . V . . . –

2 . . – – – C – . – . M – – W . – –

3 . . . – – D – . . N – . X – . . –

4 . . . . – E . O – – – Y – . –

5 . . . . . F . . – . P . – – . Z – – . .

**5. Alphabetic Telephone Number Translator**

Many companies use telephone numbers like 555-GET-FOOD so the number is easier for

their customers to remember. On a standard telephone, the alphabetic letters are mapped

to numbers in the following fashion:

A, B, and C \_ 2

D, E, and F \_ 3

G, H, and I \_ 4

J, K, and L \_ 5

M, N, and O \_ 6

P, Q, R, and S \_ 7

T, U, and V \_ 8

W, X, Y, and Z \_ 9

Write a program that asks the user to enter a 10-character telephone number in the format

XXX-XXX-XXXX. The application should display the telephone number with any alphabetic

characters that appeared in the original translated to their numeric equivalent. For

example, if the user enters 555-GET-FOOD the application should display 555-438-3663.

**6. Average Number of Words**

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

find a file named text.txt in the *Chapter 09* folder. The text that is in the file is stored

as one sentence per line. Write a program that reads the file’s contents and calculates the

average number of words per sentence.

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

**7. Character Analysis**

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

find a file named text.txt in the *Chapter 09* folder. Write a program that reads the file’s

contents and determines the following:

• The number of uppercase letters in the file

• The number of lowercase letters in the file

• The number of digits in the file

• The number of whitespace characters in the file

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

**8. Sentence Capitalizer**

Write a program with a function that accepts a string as an argument and returns a copy

of the string with the first character of each sentence capitalized. For instance, if the argument

is “hello. my name is Joe. what is your name?” the function should return the string

“Hello. My name is Joe. What is your name?” The program should let the user enter a

string and then pass it to the function. The modified string should be displayed.

**9. Vowels and Consonants**

Write a program with a function that accepts a string as an argument and returns the number

of vowels that the string contains. The application should have another function that

Programming Exercises **369**

**370** Chapter 9 More About Strings

accepts a string as an argument and returns the number of consonants that the string contains.

The application should let the user enter a string and should display the number of

vowels and the number of consonants it contains.

**10. Most Frequent Character**

Write a program that lets the user enter a string and displays the character that appears

most frequently in the string.

**11. Word Separator**

Write a program that accepts as input a sentence in which all of the words are run together

but the first character of each word is uppercase. Convert the sentence to a string in which

the words are separated by spaces and only the first word starts with an uppercase letter. For

example the string “StopAndSmellTheRoses.” would be converted to “Stop and smell the

roses.”

**12. Pig Latin**

Write a program that accepts a sentence as input and converts each word to “Pig Latin.” In

one version, to convert a word to Pig Latin you remove the first letter and place that letter at

the end of the word. Then you append the string “ay” to the word. Here is an example:

English: I SLEPT MOST OF THE NIGHT

Pig Latin: IAY LEPTSAY OSTMAY FOAY HETAY IGHTNAY