Review Questions

1. A sequence of characters enclosed within double quotation marks is a \_\_\_\_\_\_\_\_\_\_\_.

a. symbolic string

b. literal string

c. prompt

d. command

2. To create a String object, you can use the keyword \_\_\_\_\_\_\_\_\_\_\_ before the constructor call, but you are not required to use this format.

a. object

b. create

c. char

d. new

3. A String variable name is a \_\_\_\_\_\_\_\_\_\_\_.

a. reference

b. value

c. constant

d. literal

4. The term that programmers use to describe objects that cannot be changed is \_\_\_\_\_\_\_\_\_\_\_.

a. irrevocable

b. nonvolatile

c. immutable

d. stable

5. Suppose that you declare two String objects as:

String word1 = new String("happy");

String word2;

When you ask a user to enter a value for word2, if the user types “happy”, the value of word1 == word2 is \_\_\_\_\_\_\_\_\_\_\_.

a. true

b. false

c. illegal

d. unknown

When you use == with Strings, you compare their memory addresses, not their contents.

6. If you declare two String objects as:

String word1 = new String("happy");

String word2 = new String("happy");

the value of word1.equals(word2) is \_\_\_\_\_\_\_\_\_\_\_.

a. true

b. false

c. illegal

d. unknown

The equals() method returns true when the string contents are identical, including case.

7. The method that determines whether two String objects are equivalent, regardless of case, is \_\_\_\_\_\_\_\_\_\_\_.

a. equalsNoCase()

b. toUpperCase()

c. equalsIgnoreCase()

d. equals()

8. If a String is declared as:

String aStr = new String("lima bean");

then aStr.equals("Lima Bean") is \_\_\_\_\_\_\_\_\_\_\_.

a. true

b. false

c. illegal

d. unknown

The equals() method returns true when the string contents are identical, including case.

9. If you create two String objects:

String name1 = new String("Jordan");

String name2 = new String("Jore");

then name1.compareTo(name2) has a value of \_\_\_\_\_\_\_\_\_\_\_.

a. true

b. false

c. –1

d. 1

The “d” in “Jordan” is one less than the “e” in “Jore”.

10. If String myFriend = new String("Ginny");, which of the following has the value 1?

a. myFriend.compareTo("Gabby");

b. myFriend.compareTo("Gabriella");

c. myFriend.compareTo("Ghazala");

d. myFriend.compareTo("Hammie");

The value of answer a is -8 because although the “G”s are the same, the “a” in “Gabby” is 8 less than the “i” in Ginny. The value of answer b is -8 for the same reason. The value of answer c is 1 because the “h” is Ghazala is one more than “i”. The value of answer d is -1 because the “H” in “Hammie is one less than the “G” in “Ginny”.

11. If String movie = new String("West Side Story");, the value of movie.indexOf(′s′) is \_\_\_\_\_\_\_\_\_\_\_.

a. true

b. false

c. 2

d. 3

The ‘W’ is in position 0, the ‘e’ is in position 1, and the ‘s’ is in position 2.

12. The String class replace() method replaces \_\_\_\_\_\_\_\_\_\_\_.

a. a String with a character

b. one String with another String

c. one character in a String with another character

d. every occurrence of a character in a String with another character

13. The toString() method converts a(n) \_\_\_\_\_\_\_\_\_\_\_ to a String.

a. char

b. int

c. float

d. all of the above

14. Joining Strings with a ‘+’ is called \_\_\_\_\_\_\_\_\_\_\_.

a. chaining

b. concatenation

c. parsing

d. linking

15. The first position in a String \_\_\_\_\_\_\_\_\_\_\_.

a. must be alphabetic

b. must be uppercase

c. is position zero

d. is ignored by the compareTo() method

16. The method that extracts a string from within another string is \_\_\_\_\_\_\_\_\_\_\_.

a. extract()

b. parseString()

c. substring()

d. append()

17. The method parseInt() converts a(n) \_\_\_\_\_\_\_\_\_\_\_.

a. integer to a String

b. integer to a Double

c. Double to a String

d. String to an integer

18. The difference between int and Integer is \_\_\_\_\_\_\_\_\_\_\_.

a. int is a primitive type; Integer is a class

b. int is a class; Integer is a primitive type

c. nonexistent; both are primitive types

d. nonexistent; both are classes

19. For an alternative to the String class, and so that you can change a String’s contents, you can use \_\_\_\_\_\_\_\_\_\_\_.

a. char

b. StringHolder

c. StringBuilder

d. StringMerger

20. Unlike when you create a String, when you create a StringBuilder, you must use the keyword \_\_\_\_\_\_\_\_\_\_\_.

a. buffer

b. new

c. null

d. class