Computer Science 1	**Exercises 14.01-03**	**Date:**
Name:		**Period:**

1. Give a couple examples of something that involves *String Processing*.

2. Write a statement that supports the argument that “a string is a *simple data type*.”

3. Write a statement that supports the argument that “a string is a *data structure*.”

4. What is a string?

5. The characters in a string include 4 things. List them.

6. What is a *string literal*?

7. Consider this statement: **`city = "Dallas"`**
   What is the *string variable* and what is the *string literal*?

8. What is *concatenation*?

9. Which 2 operators can be used for *mathematical addition* and *string concatenation*?

10. Refer to your answer to the previous question. Why are these called *overloaded operators*?

11. Like arrays, the first index of a string is _______.

12. If a string has a length of **52** characters, what is its last index?

13. Consider this statement: **`fruit = "ORANGE"`**
    Write the code necessary to display the ‘**G**’.

14. Refer to the previous question. What would be the output of this program segment?

```
print(fruit[2:5])
print(fruit[1:])
print(fruit[:2])
```

15. Suppose, for whatever reason, you wanted a string that contains the word "**Hello**" 100 times.
Write the code that will generate this string, called **greeting**, using the word "**Hello**" only once.

16. In Python, arrays and strings can only be multiplied by __________ values.

17. What is the output of `print("Apples" == "Apples")` ?

18. What is the output of `print("Apples" == "Oranges")` ?

19. Look at program **StringOperators07.py** and its outputs.
Explain why the 4th output says "**apple goes alphabetically before ZEBRA**".

20. Suppose you have a really, really long string literal.
List 2 ways make the entire string literal visible on the computer screen.

21. What operator is used to define a *multiline string literal*?

22. Refer to your answer to the previous question.
For quite some time, we have used this same operator for ____________________________.

23. Can a *multiline string literal* be stored in a variable?

24. In the previous chapter, you learned that the **len** function returns the number of items in an array.
Does **len** work with strings? If so, what does it return?

25. Look at program **StringCommands02.py** and its output.
Both strings seem to be storing the exact same *multiline string literal*. Why do they have different lengths?

26. Can strings be *traversed*, like arrays?

27. Explain how to display a string in reverse order.

28. In the previous chapter, you learned that the **reverse** function will "reverse" the order of items in an array.
Does **reverse** work with strings?