Java Fundamentals
4-4: Strings
Practice Activities

Lesson Objectives:

- Instantiate (create) a String
- Describe what happens when a String is modified
- Use the + and += operators for concatenating Strings
- Interpret escape sequences in String literals
- Recognize the difference between a String and a primitive char data type
- Test Strings with the compareTo() and equals() method
- Describe why the == operator does not always work when testing String equality
- Use String methods length(), substring(), indexOf(), and charAt()

Vocabulary:

Identify the vocabulary word for each definition below.

Concatenation	Joining multiple String objects together.
Escape sequence	Specific characters that are preceded by a \ character. When evaluated, the special character is evaluated as a special function, such as tabs, newlines, etc.
Initialization	Assigning a value to a String object reference.
References Data Type	A data type that references the location in memory where an object is stored rather than a single, specific value.
String Manipulation Classes	Code available in the Java API to manipulate or return strings.
String	An Object type that stores sentences, words, or multiple characters.

Try It/Solve It:

1. Write three different ways to declare and instantiate a String object called “myString” and containing “abc”.

1. String myString = new String("abc");
2. String myString = "abc";
3. String myString = String.valueOf("abc");

2. Given the three String objects below, what will each of the following return?

   String s1 =“ABC”;

   String s2 = new String(“DEF”);

   String s3 = “AB” + “C”;

   a. s1.compareTo(s2); A. -3

   b. s2.equals(s3); B. false

   c. s3 == s1; C. True

   d. s2.compareTo(s3); D. False

   e. s3.equals(s1); E. True

3. Declare and instantiate two separate String objects, and then concatenate them together and assign them to a third arbitrary String object.

```
String fName = "William";
String lName = "Elizondo";

String fullName = fName + " " + lName;
```