Chapter 7 Strings

1.

```
s1 == s2  => true
s2 == s3  => false
s1.equals(s2) => true
s2.equals(s3) => true
s1.compareTo(s2) => 0
s2.compareTo(s3) => 0
s1 == s4 => true
s1.charAt(0) => W
s1.indexOf('j') => -1
s1.indexOf("to") => 8
s1.length() => 16
s1.substring(5) => me to Java!
s1.substring(5, 11) => me to
s1.toLowerCase() => welcome to java!
```

2. `String s = new String("new string");`
Answer: Correct

String s3 = s1 + s2;
Answer: Correct

String s3 = s1 - s2;
Answer: Incorrect

s1 == s2
Answer: Correct

s1 >= s2
Answer: Incorrect

s1.compareTo(s2);
Answer: Correct

int i = s1.length();
Answer: Correct

char c = s1(0);
Answer: Incorrect

char c = s1.charAt(s1.length);
Answer: Incorrect for two reasons:

- length should be length().
- It's out of bounds, even if the preceding problem is fixed.

3. Use the method `equalsIgnoreCase`.

4. Use the methods `toLowerCase` ot `toUpperCase.`
5. 0.
6. Use the overloaded static valueOf method in the String class.
7. A lowercase letter is between ‘a’ and ‘z’. You can use the static isLowerCase(char) method in the Character class to test if a character is in lowercase.
8. An alphanumeric character is between ‘0’ and ‘9’, or ‘A’ and ‘Z’, or ‘a’ and ‘z’. You can use the static isLetterOrDigit(char ch) method in the Character class to test if a character is a digit or a letter.
9. Use the StringBuffer’s constructor to create a string buffer for a string, and use the toString method in StringBuffer class to return a string from a StringBuffer.
10. StringBuffer sb = new StringBuffer(s);
    sb.reverse();
    s = sb.toString();
11. StringBuffer sb = new StringBuffer(s);
    sb.delete(4, 10);
    s = sb.toString();
12. It is actually an array.

13. StringTokenizer st = new StringTokenizer(s, "/\");

14. I
    am
    learning Java