

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.lastIndexOf('a') => 14
s1.lastIndexOf("o", 15) => 9
s1.length() => 16
s1.substring(5) => me to Java!
s1.substring(5, 11) => me to
s1.startsWith("Wel") => true
s1.endsWith("Java") => true
s1.toLowerCase() => welcome to java!
s1.toUpperCase()=> WELCOME TO JAVA!
" Welcome ".trim() => Welcome
s1.replace('o', 'T') => WelcTme tT Java!
s1.replaceAll("o", "T") => WelcTme tT Java!
s1.replaceFirst("o", "T") => WelcTme tT Java!
s1.toCharArray() returns an array of characters consisting of W, e, l, c, o, m, e, , t, o, , J, a, v, a
```

(Note that none of the operation cause the contents of a string to change)

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 : it's out of bounds, even if the preceding problem is fixed.

3. Use the method `equalsIgnoreCase`.
4. Use the methods `toUpperCase` and `toLowerCase`. The conversion methods (`toLowerCase`, `toUpperCase`, `trim`, `replace`) don't change the contents of the string that invokes these methods. These methods create new strings.
5. 0.
6. Use the overloaded static `valueOf` method in the String class.
7. The text is declared in Line 2 as a data field, but redeclared in Line 5 as a local variable. The local variable is assigned with the string passed to the constructor, but the data field is still null. In Line 10, `test.text` is null, which causes `NullPointerException` when invoking the `toLowerCase()` method.
8. The constructor is declared incorrectly. It should not have `void`.

9. 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. An uppercase letter is between 'A' and 'Z'. You can use the static `isUpperCase(char)` method in the Character class to test if a character is in uppercase.
10. 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.
11. 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`.
12.

```
StringBuffer sb = new StringBuffer(s);  
sb.reverse();  
s = sb.toString();
```
13.

```
StringBuffer sb = new StringBuffer(s);  
sb.delete(4, 10);  
s = sb.toString();
```
14. Both string and string buffer use arrays to hold characters. The array in a string is fixed once a string is created. The array in a string buffer may change if the buffer capacity is changed. To accommodate the change, a new array is created.
15.

(1) Java is fun

(2) JavaHTML

(3) Jais funva

(4) JHTMLava

(5) v

(6) 4

(7) Jav

(8) Ja

(9) avaJ

(10) JComputera

(11) av

(12) va

16. `StringTokenizer st = new StringTokenizer(s, "%$#");`

17. The total number of tokens is 2

Ja

a is cool

Any tokens left? 0

If Line 7 is replaced by the `"StringTokenizer st = new StringTokenizer(s, "/\\.\"", true);"`, the output would be

The total number of tokens is 4

Ja

v

a is cool

.

Any tokens left? 0

18.

```
String s = "Welcome to Java! Welcome to HTML! Welcome to JBuilder!";
```

```
Scanner scanner = new Scanner(s);
```

```
scanner.useDelimiter("to");
```

```
while (scanner.hasNext())
```

```
    System.out.println(scanner.next());
```

19.

```
Scanner scanner = new Scanner(System.in);  
int i = scanner.nextInt();
```

19.

(1)

Number of strings is 4

I

have

a

dream

(2)

Number of strings is 1

1 2 3

(3)

Number of strings is 0

(4)

Number of strings is 1

*

(5)

Number of strings is (the number of files and directory from where the command is executed)

Displays all files and directory names in the directory where the command is executed.