Model 1 Non-Static Methods

Method	Returns	Description
`charAt(int)`	`char`	Returns the char value at the specified index of `this` string.
`indexOf(String)`	`int`	Returns the index within `this` string of the first occurrence of the specified substring.
`length()`	`int`	Returns the length of `this` string.
`substring(int, int)`	`String`	Returns a new string that is a substring of `this` string (from `beginIndex` to `endIndex - 1`).
`toUpperCase()`	`String`	Returns a copy of `this` string with all the characters converted to upper case.

Each method listed above is non-`static`. That is, they have an ***implicit parameter*** named `this` that is passed automatically. (Note: There are many other `String` methods not listed above.)

Questions (15 min) **Start time:**

1. If the variable `str` references the string `"hello world"`, then what is the return value of the following method calls?

a) `str.charAt(8)` `'r'`

b) `str.indexOf("wo")` 6

c) `str.length()` 11

d) `str.substring(4, 7)` `"o w"`

e) `str.toUpperCase()` `"HELLO WORLD"`

2. Explain what precedes the dot operator (`.`) in the expressions above. What does it have to do with the keyword `this` in Model 1?

The variable referencing the string precedes the dot operator. It is passed as the implicit `this` parameter to the non-`static` method.

3. How many arguments does each method call in #1 have? (Hint: None of them have zero; don't forget to count the implicit argument.)

a) 2

b) 2

c) 1

d) 3

e) 1

To call a `static` method, you write *ClassName.methodName*, for example: `Math.abs(-5)`

To call a non-`static` method, you write *objectName.methodName*, for example: `str.charAt(8)`

Methods can be designed either way. Most `String` methods are non-`static`, because that makes the code easier to read.

Static (`str` passed explicitly)	**Non-Static (`str` passed implicitly)**
`String.charAt(str, 8) // wrong`	`str.charAt(8)`

4. Label each method below as `static` or non-`static` (based on how it is called).

a) `color.indexOf("RED")` non-static

b) `String.format("%3d", x)` static

c) `title.substring(0, 10)` non-static

d) `String.valueOf(3.14)` static

e) `name.charAt(0)` non-static

5. Consider the following statement and compiler error. Why is it impossible to call `charAt` this way? How would you correct the error?

```
char c = String.charAt(0);
```

Error: non-static method charAt(int) cannot be referenced from a static context

`charAt` is a non-static method; you can't call it using the `String` class. You need to call it using an object like `name`. Otherwise, the compiler won't know which string you want.

6. For each method in #4, what object is referenced by `this`? (Write N/A if `this` is not passed to the method.)

a) color

b) N/A

c) title

d) N/A

e) name