Inspiring Greatness

ISD

INDEPENDENCE SCHOOL DISTRICT

Computer Science Virtual Learning

HS Computer Science A

April 23rd, 2020

Lesson: String Substring

Objective/Learning Target:

Understanding what a substring is and how it is used in Java

What is a Substring?

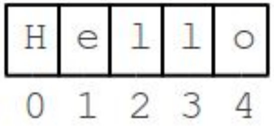
The substring() method picks out a part of string using index numbers to identify the desired part. The simplest form, **substring(int start)** takes a start index number and returns a new string made of the chars starting at that index and running through the end of the string:

```
String str = "Hello";

String a = str.substring(1);  // a is "ello" (i.e. starting at index 1)

String b = str.substring(2);  // b is "llo"

String c = str.substring(3);  // c is "lo"
```



H	e	l	l	o
0	1	2	3	4

Above str.substring(1) returns "ello", picking out the part of "Hello" which begins at index 1 (the "H" is at index 0, the "e" is at index 1).

String Index Errors: "String Index Out Of Bounds" or "String Index Out Of Range"

It is very common to get little errors with the index numbers fed into substring(). The valid index numbers for substring are basically 0, 1, 2, ... str.length(), so code needs to be careful not to pass in numbers outside that range. Note that the last number, str.length(), is one beyond the end of the string. You need this number to fit the "up to but not including" way that substring() works. For the above "Hello" examples, the valid index numbers are always in the range 0..5 since the length of "Hello" is 5.

String Index Errors: "String Index Out Of Bounds" or "String Index Out Of Range"

Often avoiding substring() out of bounds errors involves adding logic to check the length of the string. For example, suppose we want to take the first 4 chars of a string, like this…

```
// Suppose we want the first 4 chars of str
String a = str.substring(0, 4);  // WRONG error sometimes
```

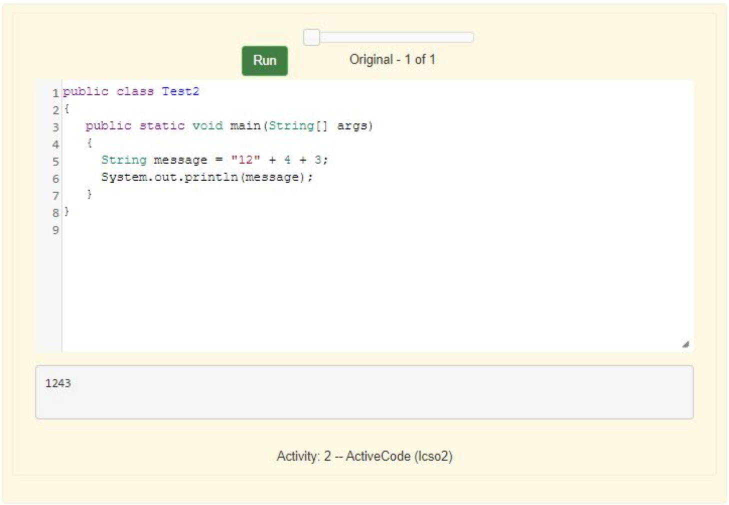
The problem with the above is .. what if the str length is less than 4? In that case, substring(0, 4) refers to non-existent chars and will fail wen run. One possible solution will add if-logic like this:

```
if (str.length() >= 4) {
  a = str.substring(0, 4);
}
else {
  // whatever you want to do when length is < 4
}
```

The point: don't assume that a string is long enough, check the length() before calling substring()

Common Mistakes

- Using `==` to test if two strings are equal. This is actually a test to see if they refer to the same object. Usually you only want to know if they have the same characters in the same order. In that case you should use `equals` or `compareTo` instead.
- Treating upper and lower case characters the same in Java. If `s1 = "Hi"` and `s2 = "hi"` then `s1.equals(s2)` is false.
- Thinking that substrings include the character at the last index when they don't.
- Thinking that strings can change when they can't. They are immutable.
- Trying to invoke a method like `indexOf` on a string reference that is null. You will get a null pointer exception.



```java
public class Test2
{
   public static void main(String[] args)
   {
     String message = "12" + 4 + 3;
     System.out.println(message);
   }
}
```

```
1243
```

Activity: 2 -- ActiveCode (lcso2)

Check Your Understanding

1. What is the value of s1 after the following code executes?

```
String s1 = "Hey";
String s2 = s1.substring(0,1);
String s3 = s2.toLowerCase();
```

a. Hey
b. he
c. H
d. h

2. What is output from the following code?

```
String s = "Georgia Tech";
String s1 = s.substring(0,7);
String s2 = s1.substring(2);
String s3 = s2.substring(0,3);
System.out.println(s3);
```

a. org
b. eor
c. eorg
d. orgi
e. You will get an index out of bounds exception

For More Resources and to Check Answers

Go to: https://runestone.academy/runestone/books/published/apcsareview/Strings/sMistakes.html

https://codingbat.com/doc/java-string-substring.html