



CS 149

Professor: Kevin Molloy



Objectives for this Week

- Learn more about working with Strings.
- Start working with loop structures.
- Work more with unit testing (JUnit).



Strings

A **String** in Java is an **object**.

Objects have methods that operate on the object.

```
String firstName = "Kevin";  
String lastName = "Molloy";  
  
String first2 = firstName.substring(1,2);  
String s = lastName.substring(0,2);  
String last2 = firstName.substring(3);
```

Offsets into a String start a 0.



String (common methods)

To determine the length of a string

int	length()	Returns the length of this string.
-----	-----------------	------------------------------------

```
String firstName = "Kevin";  
System.out.printf("Length of %s is %d\n", firstName,  
    firstName.length() );
```

To extract a single character:

char	charAt (int index)	Returns the char value at the specified index.
------	---------------------------	--

```
int i = 3;  
System.out.printf("At position %d is char %c.\n",  
    i, firstName.charAt(i) );
```



String (common methods)

To extract a portion of the String:

String	substring (int beginIndex)	Returns a string that is a substring of this string.
String	substring (int beginIndex, int endIndex)	Returns a string that is a substring of this string.

```
int i = 3;
System.out.printf("From 3rd character on is:%s\n",
    firstName.substring(2) );
System.out.printf("The ev in Kevin is:%s\n",
    firstName.substring(1,3) );
```

Notice that substring does NOT include the endIndex value (3 in the above example).



String (common methods)

To find where in a string a character is:

int	<code>indexOf(int ch)</code>	Returns the index within this string of the first occurrence of the specified character.
int	<code>indexOf(int ch, int fromIndex)</code>	Returns the index within this string of the first occurrence of the specified character, starting the search at the specified index.
int	<code>indexOf(String str)</code>	Returns the index within this string of the first occurrence of the specified substring.

```
int posOfe = firstName.indexOf('e');
```

So, what do the following do:

```
int posOfE = firstName.indexOf('E');
```

```
int posOfz = firstName.indexOf('z');
```



Strings – A Few Questions

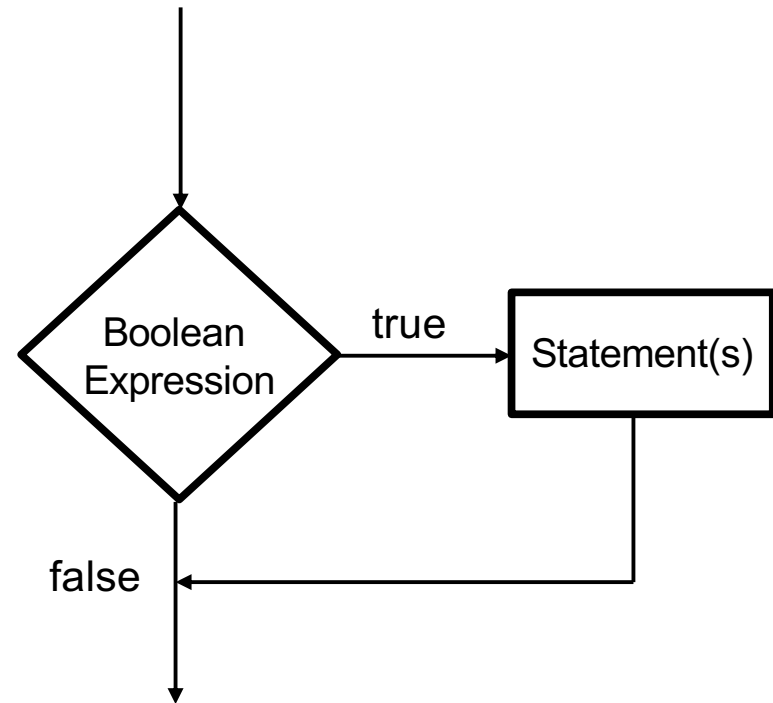
```
String work = "Hello";
```

Write JAVA statements to:

1. Calculate the length of the String work
2. Extracts the last 3 characters (only using variables)
3. Find the index of the letter "l" in the String
4. Find the index of the second letter "l" in the String.

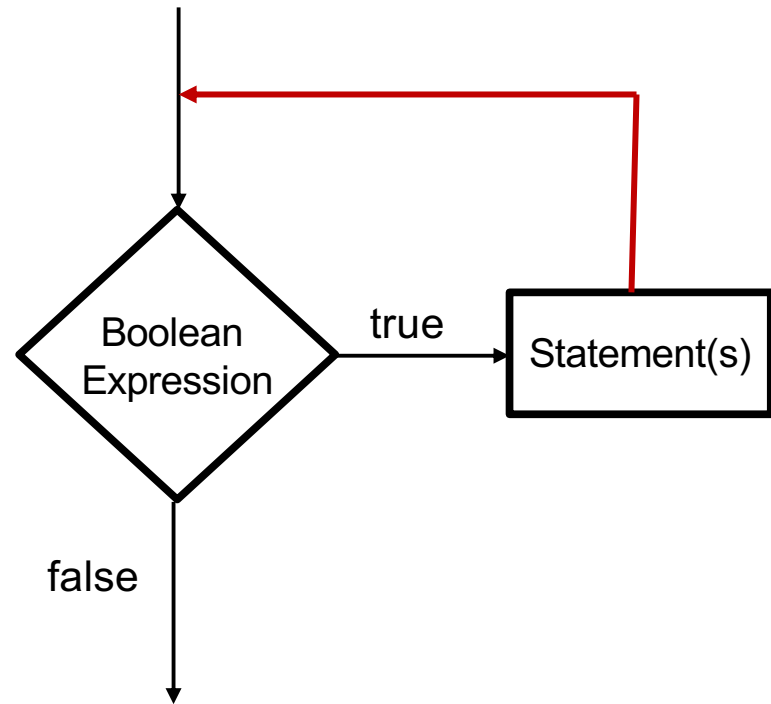
Reminder: if-statements

```
if (BooleanExpression) {  
    Statement(s)  
}
```



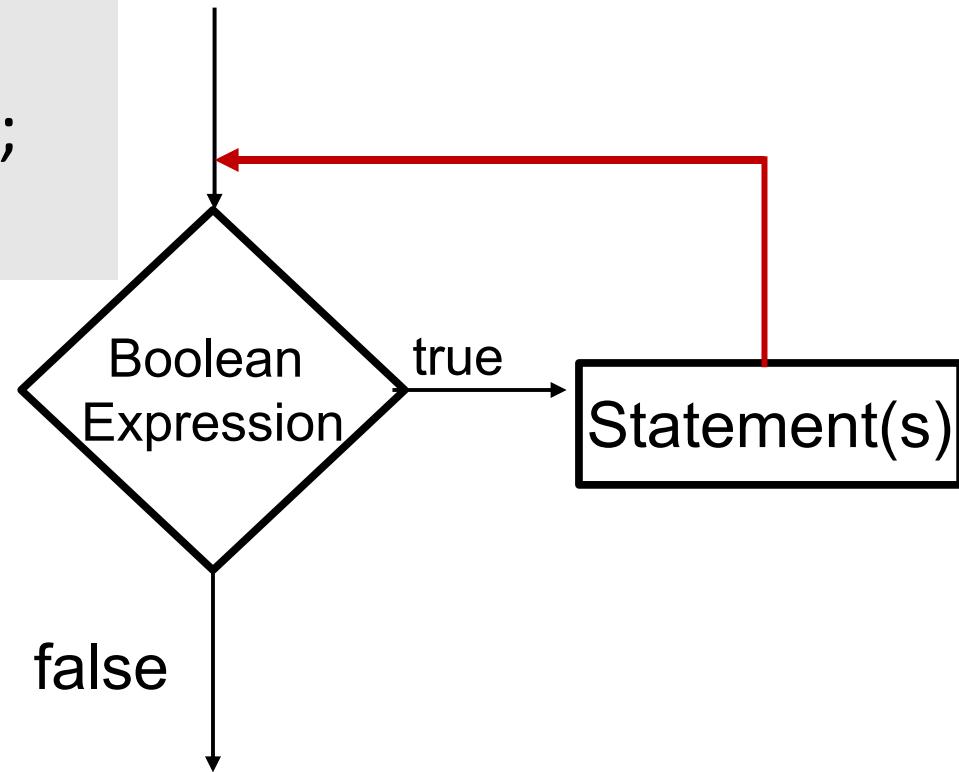
While Loops

```
while (BooleanExpression) {  
    Statement(s)  
}
```



While Loops

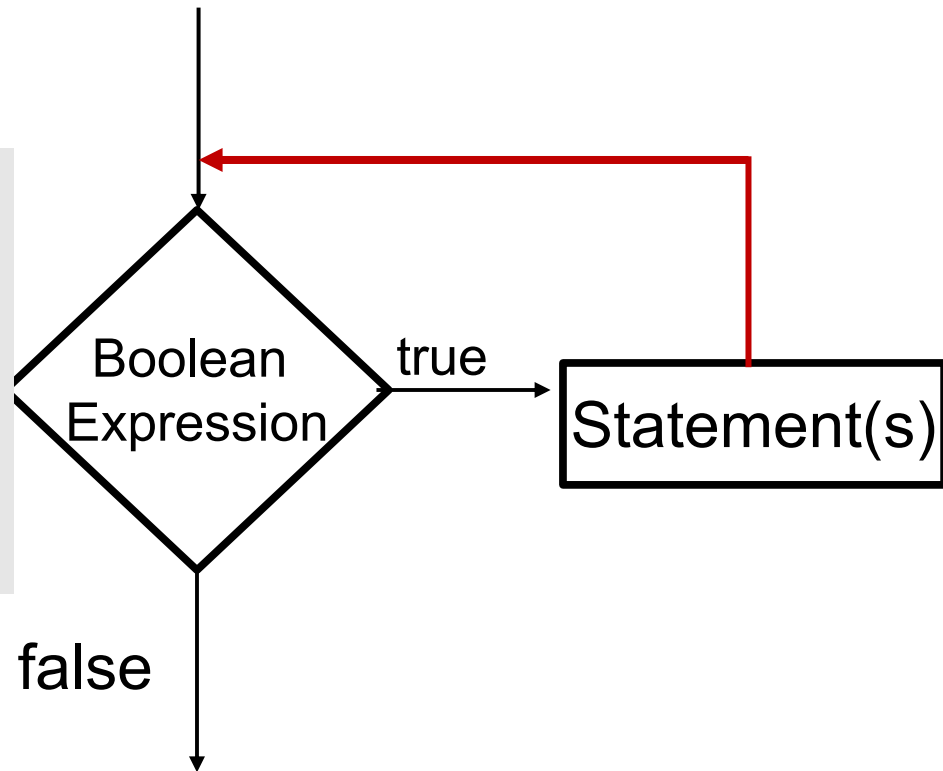
```
int a = 0;
while (a < 5) {
    System.out.println("Hello.");
}
```



Is there a problem?

While Loops

```
int a = 0;
while (a < 5) {
    System.out.println("Hello.");
    a++;
}
```



The body of every while loop should contain instruction(s) that can change the truth value of the logical expression



If for Input Validation

- We can use an if-statement to make sure that the user enters valid data:

```
System.out.print("Withdrawl amount: ");  
amount = input.nextDouble();  
  
if (amount < 1.0 || amount > 300.0) {  
    System.out.println("Bad withdrawal amount!");  
    System.exit(0); // Exits the application.  
}  
  
System.out.printf("Here are your %.2f dollars.", amount);
```

- Problem: user only gets one shot.



- **Acknowledgements**

Parts of this activity are based on materials developed by David Bernstein.

</end>