CS 312 Introduction to booleans

Jeffrey Wang

Boolean

- **boolean** is another primitive data type in Java, just like int, double, and char
- A variable of the boolean type can have one of two values: **true** or **false**
- Examples:
  - `boolean todayIsFriday = true;`
  - `boolean todayIsTuesday = false;`

The following slides are borrowed from material written by Chand John, available on the CS 312 Spring 2020 Piazza, note @529.

Review of literals and expressions

- Literals
  - 204 is of type int
  - 204.0 is of type double
  - "A" is a literal of type String
  - 'A' is a literal of type char
- Expressions
  - 3 + 4 / 5 + 5 % 6 is of type int
  - 3 + 4 / 5.0 + 5 % 6 is an expression of type double
  - 3 + "4" is an expression of type String

Boolean literals and expressions

- Only two boolean literals: true and false
- Expressions can also return boolean values
  - `3 < 4` evaluates to true
  - `4 < 3` evaluates to false
  - `"A".equals("B")` evaluates to false
  - `500 == 250 * 2` evaluates to true
  - `2 <= 200` evaluates to true

Using boolean expressions

```
for (int i = 1; i < 5; i++) {
    System.out.println(i);
}

int num = 2;
while (num <= 100) {
    num++;
}
```

Boolean expressions are bolded

Methods that return boolean

```java
public static boolean isDigit(String str) {
    for (int digit = 0; digit <= 9; digit++) {
        String digitAsStr = "" + digit;
        if (str.equals(digitAsStr)) {
            return true;
        }
    }
    return false;
}
public static boolean isEven(int number) {
    return number % 2 == 0;
}
```

Combining boolean expressions with logical operators

```java
if (num >= 0 && num <= 5) {
    System.out.println("The number you entered is between 0 and 5,
inclusive.");
}


if (str.indexOf("e") == -1 || todayIsFriday) {
    System.out.println("There is no e in the string, or, today is
Friday.");
}
```

Individual expressions are blue, logical operators in red. Together, they form compound expressions.

&& means logical and, || means logical or.

Acknowledgements

Assignment 5 tips

- You need to get familiar with String methods; they are going to be your friend
- Get familiar with parameters and return types, especially concepts such as pass-by-value
  - When you pass in an object to a method, you're not making a copy of the object; you are passing a variable that points to the same object.
  - However, once you make any changes inside the method with the object, you won't be making any changes to the object that was passed in when calling this method.
- Carefully read the definitions provided
- START EARLY!!!

Assignment 5 tips (cont.)

There are hints provided at the end of the assignment page!

Following the hints at the end of the page will guide you in completing the assignment correctly and will make your life a lot easier!

Hints:
Implement a multi-pass algorithm. This means your run through the text three times. Once to count the sentences, once to count the words, and once to count the syllables. Create a different method for each of these passes.
The String methods I used are: int length(), char charAt(int index), String toLowerCase(), int indexOf(char ch), and String concatenation. You should not need to use any other String methods. Recall you cannot use the split method from the String class or create any new Scanner objects besides the one connected to System.in.
Counting the number of words and syllables are quite similar. To count the number of words, determine when words (or vowels clusters) start. To do this, loop through the characters of the String. If a character is NOT a word delimiter (\t\n space .;:!?) and the preceding character is a word delimiter then that is the start of a word. To make this easier add a space to the start of the text the user types in.
Counting syllables is similar. Count the number of vowels that are not preceded by a vowel. Getting a lowercase version of the text can make this a little easier.
Use printf to print out the Flesch score.