

CSCI 2113.30 Lab 9

April 10 2017

Goals

1. Get more familiar with writing scala code

Activity

1. Download two files, Imperative.java and Imperative.scala.
2. Imperative.java contains two static methods, one that plays the fizzbuzz game from previous lab and another that calculates factorial of a number.
3. Imperative.scala contains that same methods and one extra: fizzbuzz, factorial and parenMatch.
4. The code in Imperative.scala does NOT follow functional programming paradigm. It is written in imperative style. (You haven't learned functional programming yet!)
5. Study the differences and similarities between Java and Scala code for fizzbuzz. One noticeable difference is that scala does not have for loops.
6. Re-implement the factorial function in scala using while loops.
7. To test your factorial code, uncomment lines 32~34 in the main function. You can run the code by typing:

```
$ scala Imperative.scala
```

8. Note the assert function. It tests a Boolean expression and throws an exception if the expression is false. If it's true, then it doesn't do anything.

Assignment

1. Implement the paranMatch function. It takes an array of characters and determines if there are matching set of parentheses.
2. Examples:

(if (zero? x) max (/ 1 x))	YES
xyz (blah (y) blah). (abc)	YES
()))(NO
hello :)	NO

3. It is not enough to just count the numbers of (and) as you can see from the examples above.
4. One way to do it is to loop through the array and count up if you encounter opening paran and count down if you encounter closing paran. If your counter ever dips into negatives, then then you have missing match. Also, if you end up with a non-zero number at the end, then you have missing match.

5. Here's how to use arrays in Scala:

```
val arr = Array('a', 'b', 'c')
println(arr(1))           //prints 'b'
arr(0) = 'x'              //sets the first element to 'x'
arr.length                //returns 3
arr(2) == 'c'             //returns true
"Hello".toArray           //Turn a string into Array of Chars
```

6. Submit the Imperative.scala file