

Lab 16

Recursion

Objectives of this Lab:

1. This lab will introduce you to recursion. You will be doing a few tasks - all using recursion - NO LOOPS ALLOWED!!
-

Overview

For this assignment you will create a Lab 16 project and class. The class will implement the following interface: [IR18.java](#)

Assignment Overview

Recursion is essentially when a method calls itself, asking for the solution of a smaller instance of the problem.

Two things are vital when using recursion: a **base case** (or base cases) and the **recursive case**.

First, we will be completing the code for creating a star pattern. There are two methods that work together here. The

```
starString(int x)
```

method is a recursive method that returns a string with the appropriate number of stars to

```
starPattern(int x)
```

to be printed. EXAMPLE OUTPUT IF X == 5:

```
*****
```

```
****
```

```
***
```

```
**
```

```
*
```

Next, we will implement the method

```
palindrome(String test)
```

that accepts a string and returns true or false depending on whether it is a palindrome or not. You can use any method of the String class that you think will be useful. Both of these solutions must be recursive and cannot use loops. Think about what the base cases are and what the recursive case is.

Below is the test code for the TA to verify.

```
public static void main(String args[]){
    R18 rec = new R18();
    rec.starPattern(5);

    System.out.println();
    System.out.println ("\'x\' is a palindrome?: " + rec.palindrome("x"));
    System.out.println("\'car\' is a palindrome?: " + rec.palindrome("car"));
    System.out.println("\'racecar\' is a palindrome?: " +
rec.palindrome("racecar"));
    System.out.println("\'hannah\' is a palindrome?: " +
rec.palindrome("hannah"));
    System.out.println("\'banana\' is a palindrome?: " +
rec.palindrome("banana") + "\n");
}
```

Turn In

Show your work to your TA before submitting to GitHub. Submit all of your files with correct headers to GitHub here: https://classroom.github.com/a/j_cgBDsJ