ISM3230 In-class lab Week 7, Module 5 – Working with Loops Fall 2015

FOCUS TOPICS

- Looping logic

TASK

Your task is to take an input String and determine whether it meets the criteria for a palindrome. For the purposes of this lab, a palindrome is a word or phrase that reads the same backward or forward.

- Your input String can consists of any characters, including spaces.
- To be a valid palindrome for this lab, the characters must match exactly reading backward or forward, including spaces and punctuation.

Some examples of words and phrases that meet this criterion:

Palindrome
151
99999
ala
madam
esse
able was i ere i saw elba
able was i ete i saw elba
radar
1234567890987654321
Oh no!no hO
Hello, worlddlrow ,olleH

Some examples of words and phrases that do not meet this criterion:

Palindrome
152
98999
alb
madama
Esse
able was i ere e saw elba
rader
1234567800987654321

As input, let the user enter a word or phrase. You can expect that the input word contains only printable characters, and contains no leading or trailing spaces.

As output, report the original word, and whether or not you found it to be a palindrome.

INSTRUCTIONS

1. Create a new NetBeans project.
2. Prompt the user to enter a word or phrase to be checked.
3. Read the input from the keyboard, and store it in a String variable.
4. Check the input to determine if it is a palindrome. This will require you to work from start to middle, and end to middle to get the characters to be compared. You should be using a loop to consider each character pair in turn.
    - **Hint**: If you pull char values from the input String, you can use == or != to check for equivalence or non-equivalence.
    - **Hint**: If you pull String values from the input String, **YOU CANNOT USE** == **TO TEST FOR EQUIVALENCE**. For String values, use the String `equals(String)` method to test for equivalence.
    - **Hint**: You will need some mechanism to keep track of whether you have found mismatched characters in the input. Consider a **boolean** typed variable.
5. Following the sample output below, report whether you found a palindrome, and report the original input string.
6. When your code is working, upload your .java file to the Week 7, Module 5 in-class lab dropbox.

SAMPLE OUTPUT

```
Enter potential palindrome:
123989321
Found palindrome: 123989321
BUILD SUCCESSFUL (total time: 9 seconds)


Enter potential palindrome:
MadaM
Found palindrome: MadaM
BUILD SUCCESSFUL (total time: 3 seconds)
```

```
Enter potential palindrome:
MaddaM
Found palindrome: MaddaM
BUILD SUCCESSFUL (total time: 3 seconds)
```

```
Enter potential palindrome:
able was i ere i saw elba
Found palindrome: able was i ere i saw elba
BUILD SUCCESSFUL (total time: 8 seconds)
```

```
Enter potential palindrome:
8998
Found palindrome: 8998
BUILD SUCCESSFUL (total time: 3 seconds)
```

```
Enter potential palindrome:
8999
Did not find palindrome: 8999
BUILD SUCCESSFUL (total time: 6 seconds)
```

```
Enter potential palindrome:
madaM
Did not find palindrome: madaM
BUILD SUCCESSFUL (total time: 3 seconds)
```

```
Enter potential palindrome:
rable was i ere i saw elba
Did not find palindrome: rable was i ere i saw elba
BUILD SUCCESSFUL (total time: 8 seconds)
```