ICE2

In this ICE you will practice writing functions and follow guidelines discussed in class.
Create the following functions and a main to test them.

Functions:

- Function called **drawStars**: this function returns nothing. It takes a non decimal number and when called will print stars equal to the value of its parameter.
    - For example drawStars(5) will generate the following output: *****
- Function called **printStatement**: this function returns nothing. It takes a string parameter and print the contents of that string.
    - For example printStatement(“This is ICE2”) will generate the following output: This is ICE2
- Function called **multiplyNumbers**: this function takes two non decimal parameters, calculates and return their product (value of their multiplication).
- Overload the **multiplyNumbers** to do the same job described in the previous step but using decimal parameters.
- A **palindrome** is a word, phrase, number, or other sequences of characters which reads the same backward or forward. Examples include "radar", "level", and "deified".
    - Write a recursive method in Java named isPalindrome that takes a String as its parameter and returns a boolean value indicating whether the string passed is a palindrome or not. The function should be case insensitive and consider only alphabetic characters (ignoring spaces, punctuation, and numbers).
    -
```
isPalindrome("Radar");       // Output: true
isPalindrome("Hello");       // Output: false
isPalindrome("A man, a plan, a canal, Panama!");  // Output: true
```
- Each of your functions **must** have docstyle function headers.

Main:

Your main should test your functions in the following order:

- Call **drawStars** to generate 15 stars.
- Call **printStatement** to generate the following message: “ICE2: Function Practice Start”

- Call **multiplyNumbers** with non decimal numbers of your choice and make sure to print the result.
- Call **multiplyNumbers** with decimal numbers of your choice and make sure to print the result.
- Call **printStatement** to generate the following message: "ICE2: Function Practice End"
- Call **drawStars** to generate 15 stars.
- Call **isPalidrome("civic")** and Call **isPalidrome("COSC1200")**

General Requirements

- Include an opening comment with your name, the name of the program, the date, and a short description.
- Follow the style guide! Use descriptive names and sensible datatypes for variables, constants, arrays, functions, etc. that follow our naming conventions. Use good spacing and make sure braces ({}) are located where they are supposed to be.
- Output messages must be meaningful. Displaying values is not enough, the user must understand what he is seeing.
- Take screen shots of your code and its output. Put them in a word document or **pdf**. The screenshots must be clear. **You can have the code and output in the same screenshot**.

Submission

- You will be responsible for submitting all the files (coding files and **screenshots**) on DC connect.
- A video demonstration is **NOT** necessary for your ICE exercises.
- A few things you must be aware of before submitting:
    - Make sure to submit your work on DC Connect before due date
    - Submit your .java files (**NOT** the full project, just the .java files you used to code your work)
    - No zip files please.
    - Up to -25% deduction if you decide to submit a zip or compressed file

    - Grades will be granted according to class coding guidelines, professionalism, output clarity and solution ingenuity.

- A working link to your completed private project on GitHub
    - Please make sure you add me (Sergio.Santilli@durhamcollege.ca) as a contributor on your project so I can obtain access to review.

- Late Submission are not excepted for In Class Exercises.

Good Luck!!