

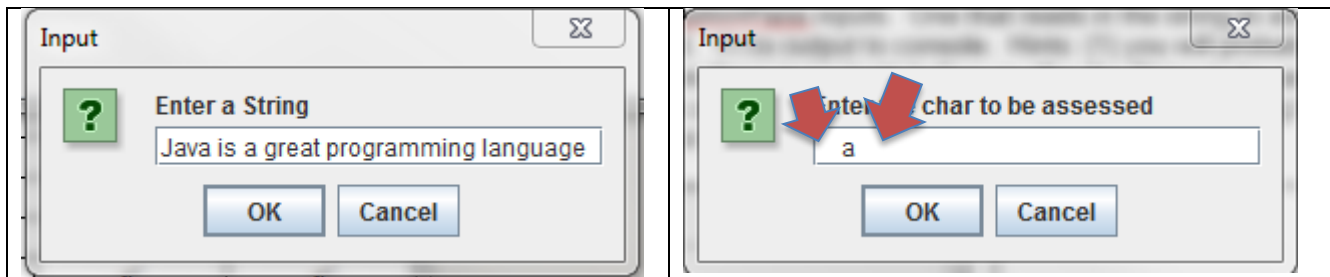
### Lab 4 (26 pts total) + 2pts XC

The purpose of this lab is to apply practical experience to Chapter 4 concepts. There are several learning objectives to this assignment

- Using loops to solve problems (for, while, do while)
- Introductory file input and output

### Extra Credit - Prog Chall 5 - pg 263 (2pts)

Your program will have two JOptionPane inputs. One that reads in the string to assess and one that reads in the letter that will be assessed. Write output to console. Hints: (1) you will probably want to trim your 2<sup>nd</sup> input used to create a char to ensure there is no 'junk' before or after the String to be assessed. (2) Remember the output of showInputDialog() is a String and you need a tool to convert from String to char. (3) This chapter is loops so you will need to use a loop to complete this. See output below.



Output:

```
In the phrase: Java is a great programming language  
There are 7 a's
```

### Temperature Problem (6pts)

You will create a program that asks the user to provide a number a space and then a letter. The number can be either an int or floating point (double). If the user enters f or F for the letter you will convert from Fahr to Cels. If the user enters c or C for the letter you will convert from Cels to Fahr. If the user enters any other letter, you will hold the user in a "while pit" until they provide a f, F, c or C. REMEMBER – The way you get into the "while pit" is the way you get out of a "while pit".

The unit input selection statement should be based on a **char-based selection statement**. Your overall program should run in a do-while loop that asks the user if they want to compute another temperature – yes or no based on a **String input selection statement**. *The easiest way to do this is create your program so that it runs once correctly and then put into a do-while loop.* Your converted temperature should be rounded to one decimal place – printf will handle this for you.

The formulas for converting temperatures are as follows:

$$^{\circ}\text{F} = ^{\circ}\text{C} \times 9/5 + 32$$

$$^{\circ}\text{C} = (^{\circ}\text{F} - 32) \times 5/9$$

Test the following inputs:

- 100 V
- Then enter F so that the input is essentially 100 F
- 212 F
- 0 C

### Output should be

- Enter C or F
- 37.8 C
- 100.0 C
- 32.0 F

(Lab 4 continued on next page)

### Reverse Word Problem (7pts)

For all the following words, if you move the first letter to the end of the word, and then spell the result backwards, you will get the original word.

banana dresser grammar potato revive uneven assess

Write a program that reads a line of words and determines whether each word possesses this property. When you get to the word quit, halt processing.

Test the following line of words: smooth jazz java potato chips quit

- (1) Do not use StringBuffer or StringBuilder reverse method to do this problem!  
(2) Type all words with one user input (ie smooth jazz java potato chips quit)

Enter words separated by a space ending with the word quit

```
smooth does not work
jazz does not work
java works
potato works
chips does not work
Enter yes to process another line?
no
```

## Flashback Ch2 Prog Chal 4 – pg 106 (8pts)

This time we will before the problems with loops. The easiest way to do this problem is to break the problem into a top half and then a bottom half. You should prompt the user for max width using JOptionPane and then write out the pattern to the console. HINT: You will want to use nested loops for both the top and bottom sections. **Also, if the user enters an even number you should add 1 to the width.** The In The Spotlight starting on pg 223 will be a big help. Recommend drawing on graph paper starting with a width of 5 to count lines, spaces, and stars to print out for the upper half and do the same for the bottom half. If you take the width that the user entered you should be able to use / and % to generate the correct vars for the nested for loops 😊 Also for the bottom half you can think of lines descending. For example, if width is 5, Top Half - line1 has 2 spaces, 1 star, line 2 has 1 space and 3 stars, and line 3 has 0 spaces and 5 stars. Bottom Half – line 2 has 1 space and 3 stars, line 1 has 2 spaces, 1 star.

Print a diamond with 8 as width, this should result in 9 stars at the max width, since 8 is an even number.

```

      *
    * * *
  * * * * *
* * * * * * *
* * * * * * *
  * * * * *
    * * *
      *

```

### Bar Chart (5pts)

Write a program that generates three (3) random integers from 0->999 and assigns to int vars. The output will have two halves. The top half will print out the random number assigned to three int vars, the bottom half will have a title NUMBER BAR CHART and for each integer, will print out one star for each hundreds. If less than 100, will print out a message, <100 no stars. See below for example output. **You must use a for loop in the bottom output portion to receive full credit.** Also, you can read ahead to Ch5 and use the printStars() in the BarChartsMethod\_Starter file or use the BarCharts\_Starter file. Closely look at each of the two Starter files to see the power of methods!

**NOTE-Based on random numbers, each student's output will be different**

```
Number 1 is: 440  
Number 2 is: 578  
Number 3 is: 36
```

```
NUMBER BAR CHART  
Number 1: ****  
Number 2: *****  
Number 3: <100 no stars
```

**Submitting your work**

For all labs you will need to provide a copy of all .java files. **No need to provide .class files. I cannot read these.** **NOTE – For Replit, please update Main.java to another name such as TempProb.java, ProChall3.java, etc.** In addition to your .java files, you will need to provide output files of your console. The name of the output file should match the class name and have the .txt extension such as TempProbOut.txt, ProChall3Output.txt. For GUIs such as JOptionPane, you will instead need to create screenshots. For Windows users, Snipping Tool is a great way to do this. Chromebook - Shift+Ctrl+Show Windows. Mac OS users, you can see how to take screenshots using the following url - <https://support.apple.com/en-us/HT201361>.