**TUTORIAL 3**

Suggested order: Complete part 1 before completing part 2,3,4

## Part 1

In this practical you should write a program that reads in a sequence of words from the user. The program should use a loop containing a `JOptionPane InputDialog` to get the next word from the user on each cycle of the loop (see Lecture notes on Iteration and Loop Statements). The program should finish when the user enters nothing (i.e. when they enter a `String` of length 0).
When the program finishes, it should use a `JOptionPane MessageDialog` to display to the user the **sum of the number of characters (the lengths)** of all the numbers entered.
For example, suppose the sequence of words entered by the user was

Hello
this
is
my
message

(where each of these `Strings` was entered in a the `JOptionPane InputDialog` on a different cycle of the loop).
The sum of the number of characters in these words is 5 + 4 + 2 + 2 + 7 = 20
Given these words the program should display the `MessageDialog`
The total number of characters in those words is 20.
Remember to finish off your program with the System.exit(0); statement (to ensure that your pop-up windows close correctly and the program ends properly).
You should name your class "WordsLength" and save it in a java file of the same name. When you have written and saved your program, you should compile it and run it using the command prompt as in practical 1 (look at part 2 of that practical for details on compiling and running programs). Make sure your program compiles and runs correctly.

## Part 2

In this practical you should write a program that reads in a sequence of words from the user. The program should use a loop containing a `JOptionPane InputDialog` to get the next word from the user on each cycle of the loop (see Lecture notes on Iteration and Loop Statements ). The program should finish when the user enters nothing (when they enter a `String` of length 0).

When the program finishes, it should first use a `JOptionPane MessageDialog` to display to the user a single sentence consisting of all the words the user entered, with a space between each word. It should then use a `JOptionPane MessageDialog` to display to the user a single sentence consisting of all the words the user entered, but in reverse order.

For example, suppose the sequence of words entered by the user was

Hello
this
is
my
message

(where each of these Strings was entered in `a JOptionPane InputDialog` on a different cycle of the loop). Given these words the program should display the `MessageDialog`

Hello this is my message

and then display the `MessageDialog`

message my is this Hello

Remember to finish off your program with the System.exit(0); statement (to ensure that your pop-up windows close correctly and the program ends properly).
When you have written and saved your program, you should compile it and run it using the command prompt as in practical 1 (look at part 2 of that practical for details on compiling and running programs). Make sure your program compiles and runs correctly.

## Part 3-Section A - StarTriangle

In this practical you will learn more about using for loops (see Lecture notes on The For Loop). Write a program which takes an integer (call it x) from the user (using a JOptionPane InputDialog) and then prints out (using the old System.out.print and println statements) a right-angled triangle of stars which is x stars high and has a base x stars long. For example, if the number x entered by the use was 5, the printed-out triangle should look like this:

```
*
**
***
****
*****
```

This program will need two for loops, one inside the other. On the first cycle of the outer loop, the inner loop will run once and print out one star; on the second cycle of the outer loop, the inner loop will run twice and print out two stars; on the third cycle of the outer loop the inner loop will run three times and print out three stars, and so on. You need to figure out how to get the inner for loop to print a different number of stars each time it runs.

Tip: start by writing a program with one for loop, which takes in a number and uses the for loop to print out that many stars. Once you have that working, put the second for loop around the first for loop and figure out the rest of the program. By the way, if you want to start a new line without printing anything on it, you can use System.out.println();.

Please call the class StarTriangle and save it in a similarly named Java file. When you have written and saved your program, you should compile it and run it using the command prompt as in practical 1 (look at part 2 of the practical for details on compiling and running programs). Make sure your program compiles and runs correctly; if any errors are produced by your program, fix them, save the file again, and recompile. Make sure you test your program with a number of different values for x and that it prints out the correct triangle for each.

## Part 3-Section B - ReverseStarTriangle

Once you have finished your program from section A, copy that program into a new file and give it a new class name ReverseStarTriangle. Then change the program so that, rather than printing a triangle which starts narrow and gets wider, it prints a triangle which starts wide and gets narrower. Your program should take an integer x from the user and print out a right-angled triangle of stars which is x stars high and has a first line that is x stars long. For example, if the number x entered by the use was 5, the printed-out triangle should look like this:

```
*****
****
***
**
*
```

When you have written and saved your program, you should compile it and run it using the command prompt as in practical 1 (look at part 2 of that practical for details on compiling and running programs). Make sure your program compiles and runs correctly; if any errors are produced by your program, fix them, save the file again, and recompile. Make sure you test your program with a number of different values of x and that it prints out the correct triangle for each.

## Part 4

Write a program which takes two ints (a `width` value and a `height` value) from the user using two `JOptionPane InputDialogs` and then prints out (using `System.out.print` and `println` statements) a box of stars with those widths and heights. The box of stars must be hollow in the center (containing blank spaces, not stars). For example, if the width value entered was 7, and the height value entered was 10, the program would print out the following box:

```
*******
*     *
*     *
*     *
*     *
*     *
*     *
*     *
*     *
*******
```

This box is 7 stars across (7 stars in the top and bottom lines, and 2 stars plus 5 spaces in the middle lines), and 10 stars high (2 top and bottom lines full of stars, plus 8 middle lines with stars at the beginning and the end). This program will need two `for` loops to draw the top and bottom lines, and will need one `for` loop inside another to draw the middle lines (one `for` loop to draw the spaces in the middle lines, and another `for` loop around that one, to do multiple middle lines). Notice that there will always be `width-2` spaces in the middle lines of a box, and 2 stars (one at the start and one at the end). Notice that there will always be one start line (full of stars) and one end line (full of stars), and `height-2` middle lines (containing 2 stars and `width-2` spaces).

Tip: start by writing three for loops, one after another: the first to do a single first line, the second to do a single middle line (with spaces), and the third to do a single last line. Once you have those working, put the second `for` loop around the `for` loop for the middle line, and figure out the rest of the program. Remember, if you want to start a new line without printing anything on it, you can use System.out.println();.

Call your class as `StarBox` and save it in an appropriately named java file.
When you have written and saved your program, you should compile it and run it using the command prompt as in practical 1 (look at part 2 of that practical for details on compiling and running programs). Make sure your program compiles and runs correctly; if any errors are produced by your program, fix them, save the file again, and recompile. Make sure you test your program with a number of different width and height numbers and that it prints out the correct box for each.