ASSIGNMENT 2

COMP-202, Winter 2015, All Sections

Due: Tuesday, February 3, 2015 (23:59)

Please read the entire pdf before starting.

You must do this assignment individually and, unless otherwise specified, you must follow all the general instructions and regulations for assignments. Graders have the discretion to deduct up to 10% of the value of this assignment for deviations from the general instructions and regulations. These regulations are posted on the course website. Be sure to read them before starting.

Question 1:	45 points
Question 2:	55 points
	100 points total

It is very important that you follow the directions as closely as possible. The directions, while perhaps tedious, are designed to make it as easy as possible for the TAs to mark the assignments by letting them run your assignment through automated tests. While these tests will not determine your entire grade, it will speed up the process significantly, which will allow the TAs to provide better feedback and not waste time on administrative details. Plus, if the TA is in a good mood while he or she is grading, then that increases the chance of them giving out partial marks. Marks can be removed if comments are missing, if the code is not well structured, and if the problems your solution does not respect the assignment requirement.

Assignment

Part 1 (0 points): Warm-up

Do **NOT** *submit this part, as it will not be graded. However, doing these exercises might help you to do the second part of the assignment, which will be graded. If you have difficulties with the questions of Part 1, then we suggest that you consult the TAs during their office hours; they can help you and work with you through the warm-up questions.*

Warm-up Question 1 (0 points)

Create a file called `Counting.java`, and in this file, declare a class called `Counting`. This class should ask the user when the computer should stop counting.

```
When should I stop counting to?
10 // User typed this
I am counting until 10: 1 2 3 4 5 6 7 8 9 10
```

Warm-up Question 2 (0 points)

For this question you have to generalize the last question. The user will give you the number they want the computer to count up to and the step size by which it will do so.

```
When should I stop counting to?
25 // User typed this
What step size should I use?
3  // User typed this
I am counting to 25 with a step size of 3:
1 4 7 10 13 16 19 21 24
```

Warm-up Question 3 (0 points)

This program prints the outline of a square made up of # signs. It should ask the user for the length of the sides in number of #s. Using two loops, you should be able to draw the outline of a square as follows.

```
How big of a square?
10     //User typed this
##########
#        #
#        #
#        #
#        #
#        #
#        #
#        #
#        #
##########
```

N.B. It is normal that the square does not appear to be a perfect square on screen as the width and the length of the characters are not equal.

How would you extend your program to output a rectangle with width and length specified by the user?

Part 2

The questions in this part of the assignment will be graded.

Question 1: Binary Converter (45 points)

You will write a program that converts binary numbers to base 10 numbers, where the input binary number is a `String`. This program will ask the user to enter a binary number using the `nextLine` method of the `Scanner` class. It will then verify whether the input only contains '0's and '1's.[1] If the user enters letters or other invalid input, the program has to keep asking the user for another input. When the input is valid, the program will then convert the binary number to a base 10 number.

Valid input - In order to check if the input is valid your program should check if it is only composed of zeros and ones. More specifically, you should scan the input string using a for or while loop to make sure it only contains '0' or '1' characters. As soon as you find a character that is not '0' or '1', the process should stop. If the process reaches the end of the input string then we have a valid input. You may ignore the case where the input string is empty, as well as any extra '0's at the start of the string.

Converting - At this point we assume that the input string is valid. The binary conversion should work as follows. For each digit in the binary number, if the digit is '1' you should add the corresponding decimal value (2^0 for the rightmost digit, 2^1 for the next digits to the left, 2^2 for the next one, and so on) to a variable that will hold the final result to be returned. This can be accomplished by using a loop.

Here is an example of what an interaction with your program should look like:

[1] **IMPORTANT:** In addition to the `nextLine` method, you are only allowed to use loops, the `length()`, `charAt(int index)` and `equals(String other)` methods of `String`s for this question.

```
Marvin: Please enter a binary number.
No!                  // This is entered by the user
Marvin: This will all end in tears. Can you just please enter a number?
Well maybe.          // This is entered by the user
Marvin: This will all end in tears. Can you just please enter a number?
101010               // This is entered by the user
Marvin: I have calculated your binary number, but you will not like it.
Marvin: The binary number 101010 is 42 in base 10.
```

Question 2: Draw a square (55 points)

You are to write a program that draws a square composed of # signs onto a grid. The user will enter the size (in the length of the sides), the x-coordinate, and the y-coordinate of the bottom-left corner of the square on the grid as command-line parameters to program, in that order. In other words, the three values will be stored in the String array `args` at the start of the program, as the values `args[0]`, `args[1]`, and `args[2]`. You can use the command `Integer.parseInt()` to convert the String into an int. Assume that the input parameters are valid ints. For example, in the Interactions pane of DrJava:

```
run Question2 5 1 1
```

means that we want to draw a square with side length 5 whose bottom-left corner is at position (1, 1).

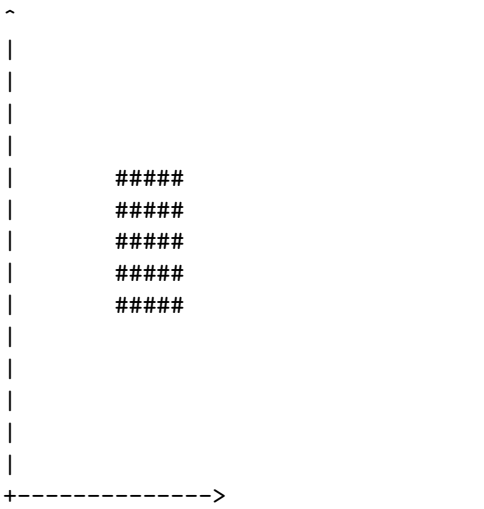
All distances in this question are in number of characters. By default, the program should print the square onto a 15x15 grid, but if the square wouldn't fit, the grid has to be extended accordingly[2]. Below are three examples:

Example 1: We have a 15 x 15 grid with a square of side length 6 at (0,0) as shown below, with input parameters of 6 0 0. Note that the first # occupies the space from coordinate (0,0) to coordinate (1,1), *etc.*

```
^
|
|
|
|
|
|
|
|
|
|
|######
|######
|######
|######
|######
|######
+--------------->
```

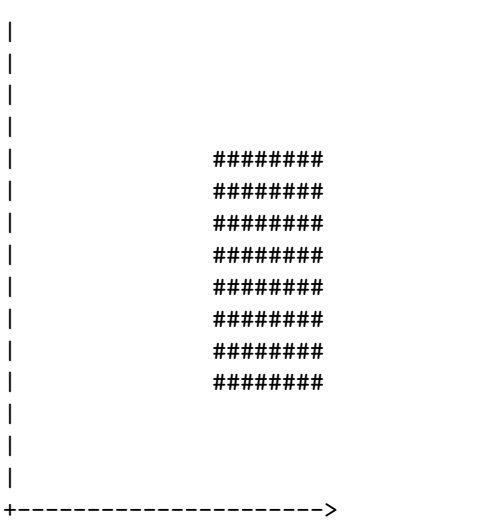
[2]Please ignore cases where the square falls into the negative, or where the side length specified is negative.

Example 2: In this case, the square is shifted 5 positions up (y) and 7 positions to the right (x), and the side length is 5 (i.e., the input parameters are 5 7 5).



Example 3:

In this case, x is 14, y is 3 and the side length is 8 (i.e., the parameter values are 8 14 3); the grid has to be extended so that the square fully fits.



The x-axis and the y-axis of the grid, and the square have to be drawn using the text output.

The program must print the axes using − for the x-axis and | for the y-axis. For the center coordinate use the + symbol. The axes should end with a $>$ for the x-axis and $\wedge$ for the y-axis.

We suggest you build your program incrementally in three steps. First, write the code to display the axes properly. Second, build a solution where the square always fits into the 15x15 grid. third, think about how you have to extend the program to make it work for larger squares or for squares that are too much shifted to fit into the 15x15 grid.

What To Submit

You have to submit one zip file that contains all your files to myCourses - Assignment 2. If you do not know how to zip files, please enquire that information from any search engine or friends. Google might be your best friend with this, and for a lot of different little problems as well.

`Question1.java` - Java code
`Question2.java` - Java code

`Confession.txt` (optional) In this file, you can tell the TA about any issues you ran into doing this assignment. If you point out an error that you know occurs in your problem, it may lead the TA to give you more partial credit. On the other hand, it also may lead the TA to notice something that otherwise he or she would not.

Marking Scheme

Up to 30% can be removed for bad indentation of your code as well as omitting comments, coding structure, or missing files. Marks will be removed as well if the class names are not respected.

Question 1

Keep asking for valid input if non-valid input entered	15	points
The input is well checked	15	points
The number is well converted	15	points
	45	**points**

Question 2

The drawn length matches the input	10	points
Update the size of the grid based on the input	10	points
The axes are drawn	10	points
x position refers to horizontal axis and y to the vertical axis	10	points
Draw a square	15	points
	55	**points**