Assignment 1

1 Objective

Practice Git, Java and Eclipse.

2 Marking

This exercise will be autotested out of of 10 marks.

It is worth 2% of your final grade.

The submission deadline is January 22, 2017 11:59pm on GitHub.

This is an individual assignment.

Late submission policy: No late submissions are accepted.

3 How to submit your work

1. Please sign in to git using your `mail.utoronto.ca` account. Use this invitiation link

   `https://classroom.github.com/assignment-invitations/9c3bb8ad2850e234f9085fb1f338c852`

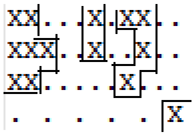
   and clone the starter code following `GitHub` instructions.
2. Complete the code for `countBlob()` method.
3. You may test your work using `BlobApp`.
4. To submit your work, add, commit and push your changes to your repository.

   Do **not** commit the files and directories generated by Eclipse, such as `bin, doc, .project`, etc. Marks will be deducted if you submit these.
5. The date and time of your last commit must be on or before 22 Jan 2017, 11:59pm.

4 What is a Blob?

In this section we explain what are blobs, how to generate a grid of them, and how to count them. This problem is often encountered in image processing. We consider an image as a rectangle made of pixels, where each pixel is turned on or off, that is each pixel can be black or white. We will model such an image by a grid a characters, where a black pixel is indicated by the presence of letter `X` in the approriate position, and a white pixel is indicated by the presence of a dot in the relevant position. An important problem in image processing is identifying and labelling connected component of pixels, or `blobs`.

By definition, a `blob` is a contiguous collection of `X`'s. For example, the grid below contains four different blobs which have been outlined.



```
XX...X.XX..
XXX..X..X..
XX.....X...
. . . . . X
```

Our goal is to create an application that given a grid, counts and reports the number of blobs. In order to enable you to test, we have provided in the starter code a blob generator.

We have represented a grid internally as a two-dimensional array of `boolean` values, with `true` indicating a blob character, and `false` indicating a non-blob character

5 Counting Blobs

In order to successfully count blobs, let's make the following observations:

- A single `X` character surrounded by non-blob characters is a blob, and, if we have a blob larger than one character, removing a blob character, what remains is again a blob. (What kind of definition is this?)
- Once we have identified all characters that do belong to a particular blob, we want to mark them so they do not get counted twice.

A possible solution is to maintain a grid of the same size as the original grid, tentatively called `marked`, which is going to be used to mark individual cells of the original grid if and only if:

- The location does exist; that is it is not outside the boundaries of the grid.
- The location does contain a blob character.
- The location has not already been visited.

Please complete the code for the `countBlob()` method as required in the starter code.