

## CO518 Assessment 1

Inspired by run-length encoding, in this assessment you will compress simple, 2 dimensional, images by representing them as commands that draw them in a simple drawing language.

You are provided two Java files, `Image.java` and `Drawing.java`, and several test image files. The test image files are stored in a text format that the `Image` class has code for reading and writing. It also has code for converting the image to a PNG file for easier viewing. The `Drawing` class contains code for reading and writing lists of drawing commands. The commands are explained in the comments of that file.

### Task 1

(5 marks)

Implement the `draw` method in the `Drawing` class to create and return an image by executing all of the drawing commands in the `commands` field. Throw a `BadCommand` exception if any command tries to paint outside of the picture's dimensions, as given by the `height` and `width` fields. The position can leave the image's dimensions, as long it no attempt is made to paint outside of the image.

### Task 2.1

(6 marks)

Implement the `compress` method in the `Image` class to create and return a list of drawing commands (specifically, a `Drawing` object) that will draw this image.

### Task 2.2

(5 marks)

The `compress` method can use the ideas from run-length encoding to create a much shorter list of drawing commands. You should be able to compress the first two tests into 20 commands, for `test-image1` and 35 commands for `test-image2` for full marks. (It should be technically possible to get down to 14 and 29.) (You can to hand-craft a list of commands to explore the problem, but this task requires that your compression algorithm generates the commands.)

## Task 2.3

(4 marks)

We will have a competition to award the final 4 marks, based on how well you compress all of the test images. For some images it might be possible to do much better than a simple run-length encoding, and you encouraged to be creative. For each image, we will rank the correct solutions by size, from 1st place for the smallest down to 200-ish place (depending on the number of correct solutions for that image). Then we will add your ranks across all of the images. The lowest summed-rank is the winner. The lowest 20% get 4 marks, the next 20% get 3 marks and so on.

## Submission

The deadline is Friday, 27 October, 2017 at 23:55. Submission is via the CO518 Moodle page. No late submissions will be accepted. Submit a single .zip file containing your updated **Drawing.java** and **Image.java** files along with files of drawing commands for each test image and pixel art. These must be in file named test-drawingN for each image file test-imageN, and pixel-art-drawingN for each image file pixel-artN. The drawing files must conform to the format specified in Drawing.java.