

Due: Friday, May 29th, before 6pm

This is a pair programming problem! You are expected to work with the person you have been paired with in class, and you are both responsible for submitting the exact same code to T-Square.

For your next task at Adobe, your boss has asked you to create a subclass of the Picture class called BetterPicture.java and add some additional functionality to the new class. You are to create methods to generate proof sheets as well as a method to crop an image. The details of each method are explained below.

Proof Sheets:

You need two static methods with the same name (proofSheet) that produce proof sheets, [These two methods share the same name, but have different types of parameters, so the name is overloaded.] One version takes in an array of Pictures and one takes in a single Picture. The syntax should read:

```
Picture canvas = BetterPicture.proofSheet(Picture pic)
```

and:

```
Picture canvas = BetterPicture.proofSheet( Picture [ ] picArray )
```

- Each method should return a (newly created) 700x700 image.
- The proof sheet should contain up to four images scaled to fit in a 350x350 quadrant
 - i.e. the largest side of any resulting picture should be 350 pixels
 - Aspect ratio of pictures must be retained!
- The images must be centered in their quadrant
 - i.e. if an image is 300 pixels tall, it should have 25 pixels of space on the top and bottom
- The method with a single picture parameter should compose from left to right then top to bottom:

- Original
- Black and White
- Negated
- Flipped
- The method with an array of pictures as the parameter should compose up to the first four images.
 - Even if the array only has 2 or 3 pictures, it should still compose them and leave some quadrants blank
 - The method should ignore any pictures after the fourth in the array
- If an array is passed in with only a single Picture element, it should call the other method on that picture.



Single Image



Multiple Images

Crop:

You need a single method for crop which takes in four integer parameters to describe where to crop and returns a cropped version of the Picture. The syntax should read:

Picture cropped = BetterPicture.crop(0,0,150,150)

- The parameters should read (xStart, yStart, xEnd, yEnd)
- Your code should test for bad input
 - i.e. xStart should be less than xEnd, all of the input should be within the picture, etc.
 - If the input is bad, the method should return null
- The picture that your function returns should be “cropped” to only contain pixels between the starting and ending points.

Rubric

- Overall
 - BetterPicture subclasses Picture correctly 5pts
- Proof Sheet
 - Headers and overloading correct 5pts
 - Images scaled correctly 15pts
 - Images composed to respective quadrants 10pts
 - Centered correctly 10pts
 - Picture array input cases handled correctly
 - 1 Picture 5pts
 - 2-4 Pictures 10pts
 - 5+ Pictures 5pts
 - Single picture proof manipulations correct 5pts
- Crop
 - Rejects bad input 10pts
 - Crops correctly 20pts