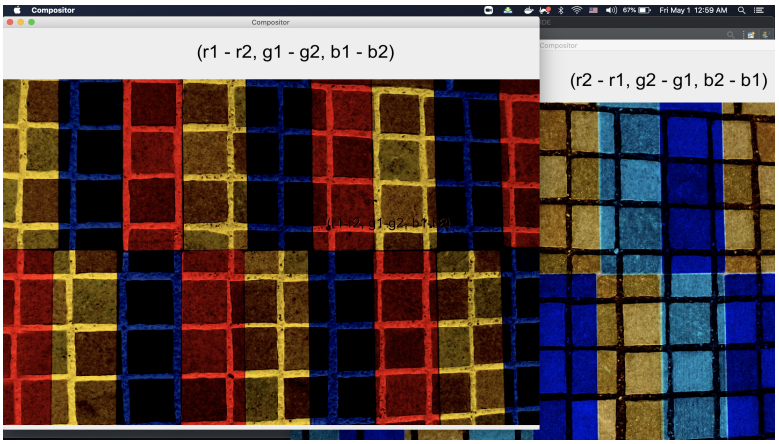
```
import java.awt.*;
import java.awt.image.BufferedImage;
import com.programwithjava.basic.DrawingKit;

public class Compositor {

  // this method returns a new image created by adding
  // the source images image1 and image2
  public BufferedImage subtract(BufferedImage image1,BufferedImage image2) {
    int width = Math.min(image1.getWidth(), image2.getWidth());
    int height = Math.min(image1.getHeight(), image2.getHeight());

    // create a new BufferedImage called image3
    BufferedImage image3 = new BufferedImage(width, height, BufferedImage.TYPE_INT_ARGB);

    for (int x = 0; x < width; x++) {
      for (int y = 0; y < height; y++) {
        // get the samples of the pixel at (x, y) in image1
        int colorValue1 = image1.getRGB(x, y);
        Color pixelColor1 = new Color(colorValue1);
        int red1 =  pixelColor1.getRed() ;
        int green1 =  pixelColor1.getGreen() ;
        int blue1 =  pixelColor1.getBlue();

        // get the samples of the pixel at (x, y) in image2
        int colorValue2 = image2.getRGB(x, y);
        Color pixelColor2 = new Color(colorValue2);
        int red2 =  pixelColor2.getRed() ;
        int green2 =  pixelColor2.getGreen() ;
        int blue2 =  pixelColor2.getBlue();

        // add the samples to create a new color
        int red3 =  Math.max(0, red1 - red2);
        int green3 =  Math.max(0, green1 - green2);
        int blue3 =  Math.max(0, blue1 - blue2);

        // set the color of a pixel in image3
        Color newPixelColor = new Color(red3, green3, blue3);
        int newRgbvalue = newPixelColor.getRGB();
        image3.setRGB(x, y, newRgbvalue);
      }
    }
    // returns a reference to the new image
    return image3;
  }

  public static void main(String[] args) {
    DrawingKit dk = new DrawingKit("Compositor", 1000, 1000);
    BufferedImage p1 = dk.loadPicture("image/pattern1.jpg");
    BufferedImage p2 = dk.loadPicture("image/pattern2.jpg");
    Compositor c = new Compositor();
    BufferedImage p3 = c.subtract(p1,p2);
    dk.drawPicture(p3, 0, 100);
  }
}
```



The result of subtracting the first image from the second is NOT the same as subtracting the second from the first.