

Step Into Java:
Collages! (not 2 b confused
w/ Colleges)

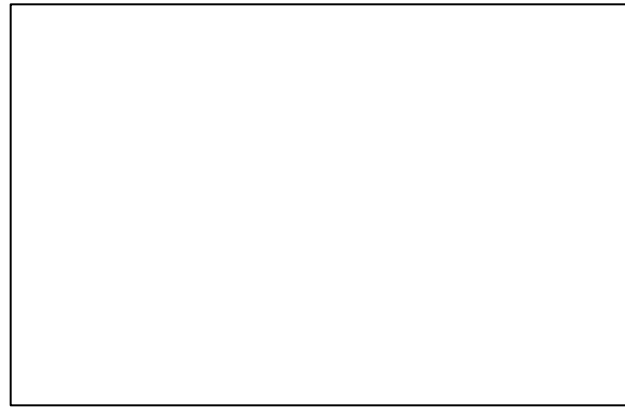
Mr. Neat
Java

Collages

- Copy one Pix to another...



flower



joe

Picture joe = new Picture(...

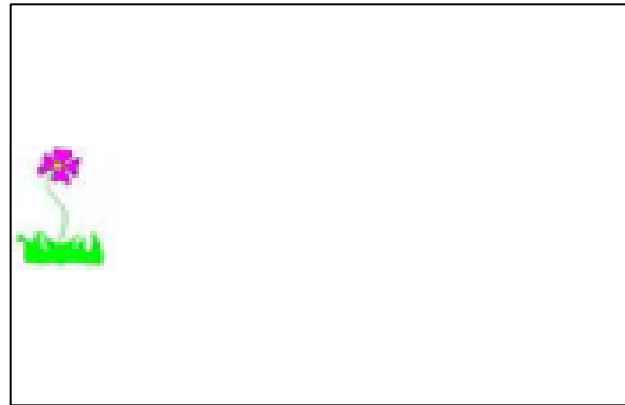
Picture flower = new Picture(...

Collages

- Copy one Pix to another...



flower



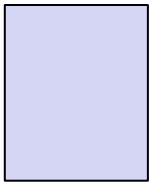
joe

Picture joe = new Picture(...

Picture flower = new Picture(...

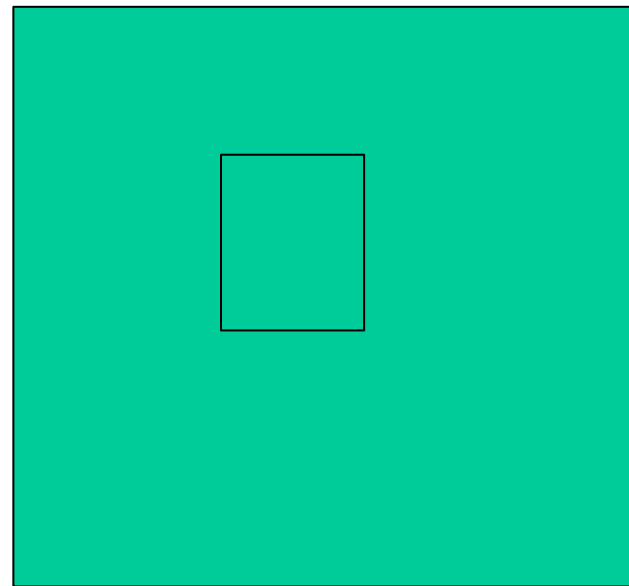
joe.copy(flower,100,0);

From Picture



flower

To Picture



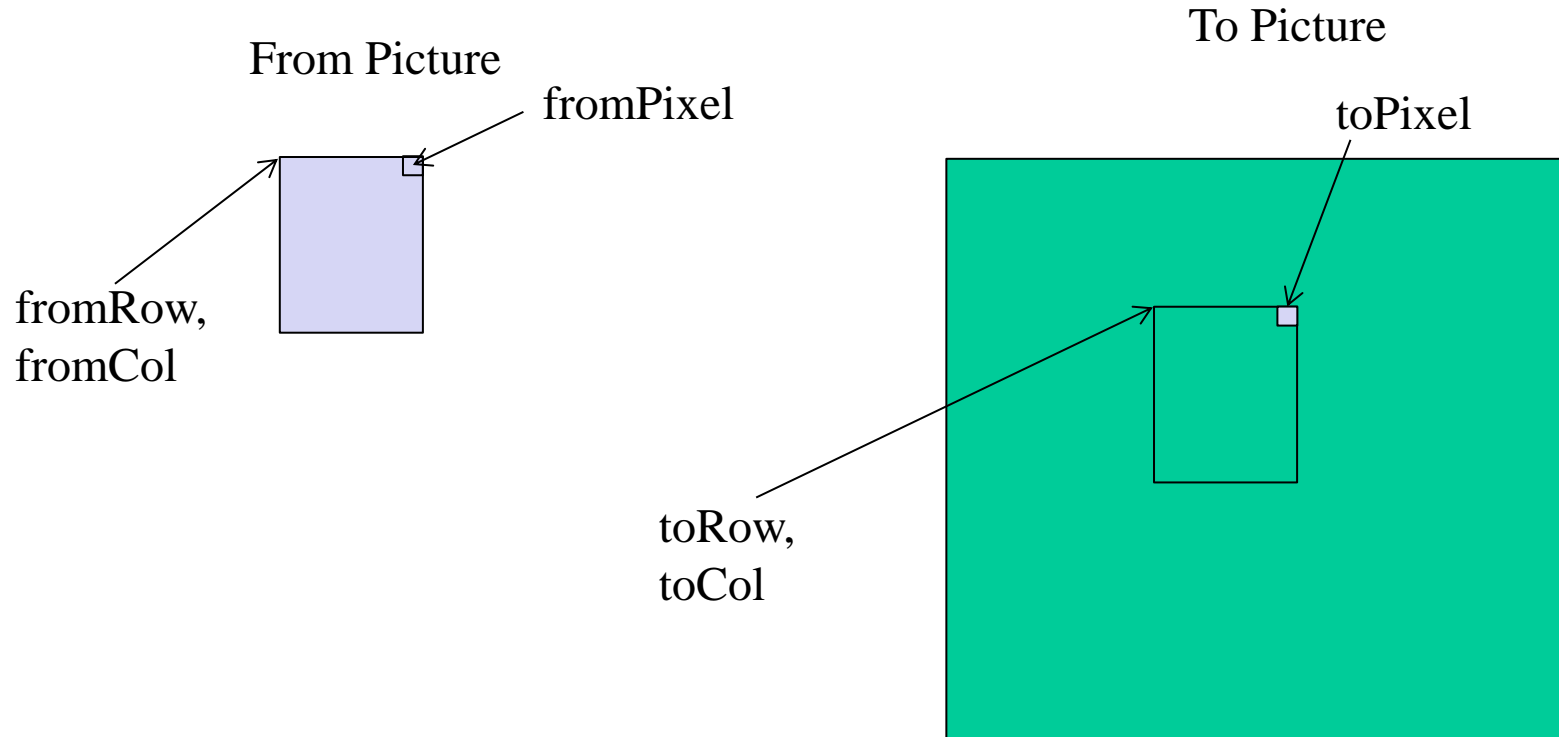
joe

```
joe.copy(flower, 100, 100);
```

Collages

- Let's explore the copy method (from the Picture class)

```
public void copy(Picture fromPic,
                int startRow, int startCol)
{
    Pixel fromPixel = null;
    Pixel toPixel = null;
    Pixel[][] toPixels = this.getPixels2D();
    Pixel[][] fromPixels = fromPic.getPixels2D();
    for (int fromRow = 0, toRow = startRow;
         fromRow < fromPixels.length &&
         toRow < toPixels.length;
         fromRow++, toRow++)
    {
        for (int fromCol = 0, toCol = startCol;
             fromCol < fromPixels[0].length &&
             toCol < toPixels[0].length;
             fromCol++, toCol++)
        {
            fromPixel = fromPixels[fromRow][fromCol];
            toPixel = toPixels[toRow][toCol];
            toPixel.setColor(fromPixel.getColor());
        }
    }
}
```



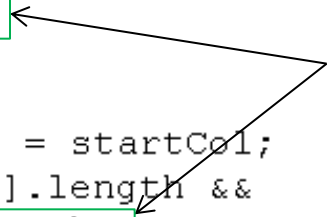
```
fromPixel = fromPixels[fromRow][fromCol];  
toPixel = toPixels[toRow][toCol];  
toPixel.setColor(fromPixel.getColor());
```

Collages

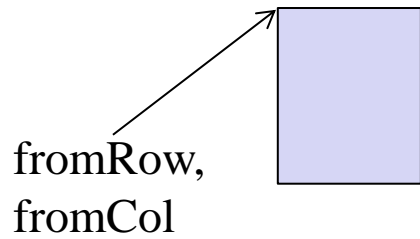
- Let's explore the copy method (from the Picture class)

```
public void copy(Picture fromPic,
                int startRow, int startCol)
{
    Pixel fromPixel = null;
    Pixel toPixel = null;
    Pixel[][] toPixels = this.getPixels2D();
    Pixel[][] fromPixels = fromPic.getPixels2D();
    for (int fromRow = 0, toRow = startRow;
         fromRow < fromPixels.length &&
         toRow < toPixels.length;
         fromRow++, toRow++)
    {
        for (int fromCol = 0, toCol = startCol;
             fromCol < fromPixels[0].length &&
             toCol < toPixels[0].length;
             fromCol++, toCol++)
        {
            fromPixel = fromPixels[fromRow][fromCol];
            toPixel = toPixels[toRow][toCol];
            toPixel.setColor(fromPixel.getColor());
        }
    }
}
```

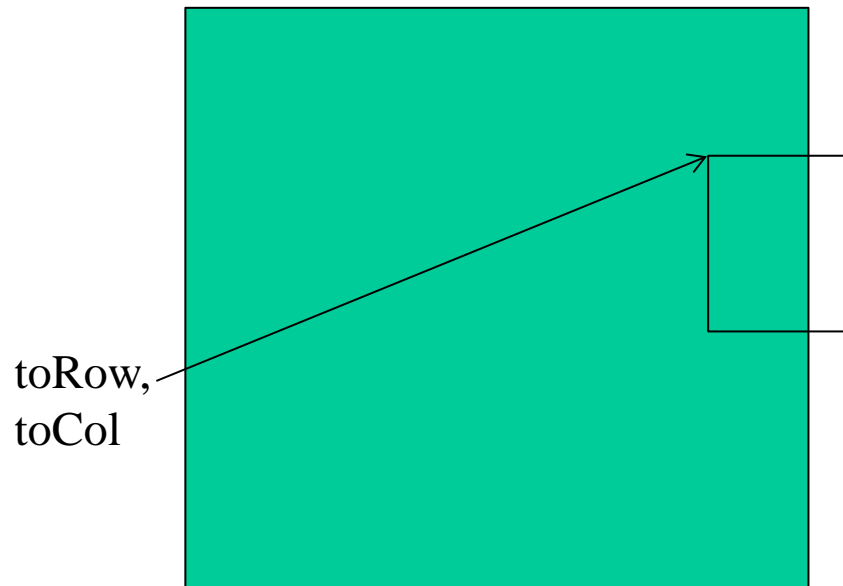
makes sure copied pix
doesn't go off page

Two arrows originate from the text 'makes sure copied pix doesn't go off page'. One arrow points to the condition 'toRow < toPixels.length;' in the first for loop. The other arrow points to the condition 'toCol < toPixels[0].length;' in the second for loop.

From Picture



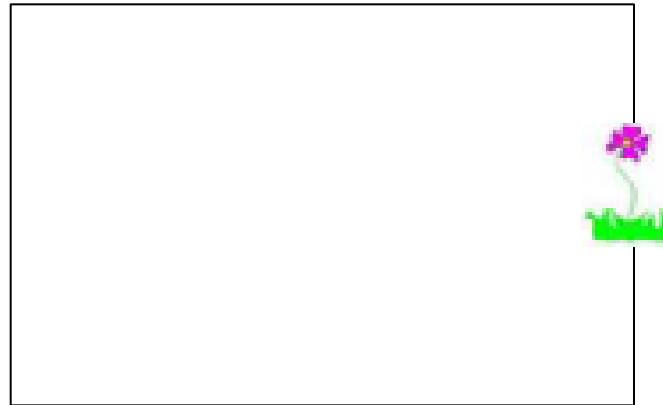
To Picture



```
for (int fromRow = 0, toRow = startRow;  
    fromRow < fromPixels.length &&  
    toRow < toPixels.length;  
    fromRow++, toRow++)
```


Collages

- Copy one Pix to another...



Collages Go Crazy!



Collages Go Crazy!

```
public void createCollage()  
{  
    Picture flower1 = new Picture("flower1.jpg");  
    Picture flower2 = new Picture("flower2.jpg");  
    this.copy(flower1, 0, 0);  
    this.copy(flower2, 100, 0);  
    this.copy(flower1, 200, 0);  
    Picture flowerNoBlue = new Picture(flower2);  
    flowerNoBlue.zeroBlue();  
    this.copy(flowerNoBlue, 300, 0);  
    this.copy(flower1, 400, 0);  
    this.copy(flower2, 500, 0);  
    this.mirrorVertical();  
    this.write("collage.jpg");  
}
```



Lab – Collage

Create a `myCollage` method that has at least three pictures (can be the same picture) copied three times with three different picture manipulations and at least one mirroring. Write a class (static) test method in `PictureTester` to test this new method and call it in the `main` method.