Spring 2012 CSI201 Lab08: Midterm exam programming makeup OR bonus.

(Write a method AND NUMBER with 1, 2, 3 each thing you write!) Your job is to add a method to the Picture class that will put centered black crosshairs into the Picture referred to by this. When myPict.crossHairs() is called, your code should change the completely white Picture labelled BEFORE to the one labelled AFTER.

BEFORE                         AFTER

In order to do this, the computer must make several computations and it must be controlled by loops to repeat certain operations. The things you must program are listed AND NUMBERED here:

1. Get the Picture’s height (with getHeight() ).
2. Get the Picture’s width (with getWidth() ).
3. Compute half the height (approximately).
4. Compute half the width (approximately).

Program two separate loops, NOT NESTED!
5. One for loop to blacken the correct row of Pixels. (The horizontal row or line should be only 1 Pixel wide.)
6. One for loop to blacken the correct column of Pixels. (The vertical column or line should be only 1 Pixel wide.)
7. Get one single Pixel pix of the horizontal row crosshair.
8. Get one single Pixel pix of the vertical column crosshair.
9. Actually blacken one Pixel in the horizontal row crosshair (use pix.setColor( Color.black ); or equivalent.
10. Actually blacken one Pixel in the vertical column crosshair.

Write Java code into the method, as complete and correct as you can make it; AND (for 20 of the 30 points!!!) NUMBER WITH (1), (2), .... (10) exactly each piece you wrote with the NUMBER of the item above that best describes its purpose. This exam question is for you prove that you understand the purpose of everything you write!!

Fill in the body of the crossHairs method on the next page, MAKE SURE YOU WRITE "(1)" , "(2)" , "(3)" , "(4)" , "(5)" , "(6)" , "(7)" , "(8)" , "(9)" and "(10)" , AND clearly mark what (1), (2) , ... (10) refer to.

BONUS PROBLEM: IF you did the whole problem during the exam, do the following bonus problem:
Add to G&E’s Picture class a method named DrawAnX and submit a Picture.java file that demonstrates your work is correct when we run it.
Level 1: void DrawAnX(int xCenter, int yCenter, int squareSideLength) 
Draws the X inside a square with given center and given length of each side.

Level 2: void DrawAnX(int Xcenter, int Ycenter, int width, int height) 
Draws the X inside a rectangle with given center and given lengths of its two sides. The lines must go from corner to corner, crossing near the middle. This takes a lot more math than the equal-side case!
Lab 08 Worksheet

Make a new Lab08 directory/folder and put a copy of Picture.java in it.

Method body headers: Put these in your Picture.java copy:

Full Picture version (from the exam)

```java
public void crossHairs()
{
}
```

Parameterized version (new for lab)

```java
public void crossHairs(int xCenter, int yCenter, int width, int height)
{
}
```

Code the loop for getting at the Pixels in the **vertical** line. Make the loop body empty at first.

Refer to textbook or course notes for a single for loop that makes int y range over the 0 to this.getHeight()-1 inclusively: A for (...;...;...) { ... } loop is best for this. Your code will eventually get at one Pixel at a time, so a reminder of some code to do that is given below. Answer to yourself: What should the value of x be? Should that value be constant or vary from one body repetition to another?

```java
public void crossHairs()
{
    //You'll have to declare and define x and y!
    //Pixel pixToChange = this.getPixel(x, y);
}
```

Code the loop for getting at the Pixels in the **vertical** line. Make the body empty. Refer to textbook or course notes for a single for loop that makes int y range over the yCenter-height/2 to yCenter+height/2 (approximately). Answer to yourself: What should the value of x be? Should that value be constant or vary from one body repetition to another?

```java
public void crossHairs(int xCenter, int yCenter, int width, int height)
{
    //You'll have to declare and define x and y!
    //Pixel pixToChange = this.getPixel(x, y);
}
```

Now add the code inside the loop body to blacken the Pixel that the computer got at.

Do this for the whole Picture version AND the parameterized version.
Reminder: pixToChange.setColor(java.awt.Color.black);
**Underneath** the whole loop (that means AFTER THE BODY!!) for drawing the vertical line, start coding the loop for getting at the Pixels in the **horizontal** line. Make the body empty at first. Then, add the same old code for blackening one Pixel. Doing the horizontal line is conceptually the same as the loop you finished, except the x and y dimensions are reversed. Answer to yourself: What should the value of y be? Should that value be constant or vary from one body repetition to another?

```java
public void crossHairs()
{
    //Imagine that here is the code to draw the vertical line.
    //...
    //The computer thinks: Wow, I drew a vertical line!
    //Start the code to draw the horizontal line below:
}
```

Now do it for the parametrized version. Hint: x range is xCenter-width/2 to xCenter+width/2, approximately. Answer to yourself: What should the value of y be? Should that value be constant or vary from one body repetition to another?

```java
public void crossHairs(int xCenter,int yCenter,int width,int height)
{
    //Imagine that here is the code to draw the vertical line.
    //...
    //The computer thinks: Wow, I drew a vertical line!
    //Start the code to draw the horizontal line below:
}
```