

Practice Exam 1

1. (4 points) In the additive RGB color system, red and green combine to make yellow, green and blue combine to make cyan, and blue and red combine to make magenta. Equal amounts of all three combine to make white. This corresponds to how the red, green, and blue receptors in the human eye respond to incoming light. By contrast, the subtractive CMY color system corresponds to mixing inks, where the color we see is only the light which is not absorbed. For example, if a certain ink absorbs blue light, then our eyes will respond to the remaining red and green light and see yellow.

(a) What color of light is absorbed by magenta ink?

(b) Mixing several inks will cause each of their corresponding colors to be absorbed. Which two inks (out of cyan, magenta, and yellow) should be mixed to make red?

(c) What color results from mixing equal quantities of all three inks, cyan, magenta, and yellow?

2. (2 points) What values would you need to put in for the variables `a` and `b` so that this loop will print the list of numbers 5 6 7 8 9 ?

```
for (int i = a; i < b; i++) {  
    System.out.print(i + " ");  
}
```

3. (10 points) Complete the following method for Java to examine each pixel of the Picture and apply the following color changing rules:

- Reduce the red component by a factor of 2;
- Increase the green component by 50% (for example, 100 should become 150)—don't worry about going over 255.

(a)

```
public void colorChange() {  
    Pixel[ ] pixs = this.getPixels();
```

```
}
```

(b) Give two examples of colors (in terms of their RGB components) which will not be changed by the above rules.

(c) Describe the effect of the `colorChange()` method in words.

4. (4 points) Describe the effect of executing `barb.modify()` (assume that `barb` is our favorite picture of Barbara Ericson, and the following code has been added to the `Picture` class):

```
public void modify() {
    width = this.getWidth();
    height = this.getHeight();
    for (int x = 0; x < width/2; x++) {
        for (int y = 0; y < height/2; y++) {
            Pixel pixelA = this.getPixel(x, y);
            Pixel pixelB = this.getPixel(x, height - y - 1);
            Color colorA = pixelA.getColor();
            Color colorB = pixelB.getColor();
            pixelA.setColor(colorB);
            pixelB.setColor(colorA);
        }
    }
}
```

5. (4 points) For each of the following program fragments, determine what will be printed:

(a)

```
for (int number = 1; number < 5; number++) {
    System.out.println(number);
}
```

(b)

```
int number = 4;
while (number > 1) {
    System.out.println(number);
    number = number - 1;
}
```

6. (4 points) Describe the effect of executing `barbCopies()`:

```
public Picture barbCopies() {
    Picture barb = FileChooser.getMediaPath("barbara.jpg");
    int width = barb.getWidth();
    int height = barb.getHeight();
    Picture canvas = new Picture(width, height);
    copy(barb, canvas, width, height, 0, 0); // upper-left
    negative(barb);
    copy(barb, canvas, width, height, width/2, 0); // upper-right
    flip(barb, width, height);
    copy(barb, canvas, width, height, 0, height/2); // lower-left
    negative(barb);
    copy(barb, canvas, width, height, width/2, height/2); // lower-right
    return canvas;
}

// Put a half-size copy of barb into canvas with upper-left corner at (tx, ty)
public void copy(Picture barb, Picture canvas, int w, int h, int tx, int ty) {
    for (int x = 0; x < w/2; x++) {
        for (int y = 0; y < h/2; y++) {
            Pixel pixelB = barb.getPixel(2 * x, 2 * y);
            Pixel pixelC = canvas.getPixel(tx + x, ty + y);
            pixelC.setColor(pixelB.getColor());
        }
    }
}

public void negative(Picture p) {
    for (Pixel pix : p.getPixels()) {
        int r = pix.getRed();
        int g = pix.getGreen();
        int b = pix.getBlue();
        pix.setColor(new Color(255 - r, 255 - g, 255 - b));
    }
}

public void flip(Picture p, int width, int height) {
    for (int x = 0; x < width/2; x++) {
        for (int y = 0; y < height; y++) {
            Pixel pixelA = p.getPixel(x, y);
            Pixel pixelB = p.getPixel(width - x - 1, y);
            Color colorA = pixelA.getColor();
            Color colorB = pixelB.getColor();
            pixelA.setColor(colorB);
            pixelB.setColor(colorA);
        }
    }
}
```