How was the exam?
A. No problem!
B. Not *too* bad
C. Just OK
D. Pretty bad
E. Horrible!
CSE 8A Lecture 9

• Reading for next class: 6.1-6.4 (conditional statements)

• PSA4 interview due tomorrow noon.

• PSA5: Collage and Picture Flip, DON’T WAIT (it’s longer than the previous PSAs)
Challenge: What does this code do?

• Hint: trace some of the getPixel index values.

```java
int magic = getWidth() / 2;
Pixel foo, bar;
for (int y = 0; y < getHeight(); y++)
{
    int countingDown = getWidth() - 1;

    for (int x = 0; x < magic; x++)
    {
        foo = getPixel(x, y);
        bar = getPixel(countingDown, y);
        bar.setColor(foo.getColor());
        countingDown--;
    }
}
```

A. Copies top half into bottom half not mirrored.

B. Copies left half into right half not mirrored.

C. Mirrors around vertical axis, left into right

D. Mirrors around horizontal axis, top into bottom

E. Some other bizarre transformation
Challenge: What does this code do?

• Hint: trace some of the getPixel index values.

int magic = getWidth()/2;
Pixel foo, bar;
for(int y = 0; y < getHeight(); y++)
{
    int countingDown = getWidth()-1;
    for(int x = 0; x < magic; x++)
    {
        foo = getPixel(x,y);
        bar = getPixel(countingDown,y);
        bar.setColor(foo.getColor());
        countingDown--;
    }
}

1) Solo: (1 min)
2) Discuss: (2 min)
3) Group: (30 sec)
By what variable name do we refer to `collage` inside `makeC` in `Picture.java`?

A. `collage`
B. `callingObject`
C. `Object`
D. `Picture`
E. `this`
Match the scenario to the constructor call (we’ll vote for each scenario)

**Scenario**

1) Create a picture from a specific file

2) Create a picture that is a copy of another picture

3) Create a picture of a given width and height

4) Create a picture of the same width and height as another picture

**Call**

A. Picture p = new Picture();

B. Picture p = new Picture("filename.jpg");

C. Picture p = new Picture(other);
   int aNum = other.getWidth();

D. Picture p = new Picture(aNum, bNum);
   int bNum = other.getHeight();
What does this code do?

```
Pixel foo;
for(int y = 40; y < 50; y++)
{
    for(int x = 1 ; x < 5; x++)
    {
        foo = getPixel(x,y);
        foo.setColor(Color.RED);
    }
}
```

Makes red box of

width  height
A.  10  5
B.  9  4
C.  5  10
D.  4  9
E. None of the above
What does this code do?

```
Pixel foo;
for(int y = 40; y < 50; y++)
{
    for(int x = 1; x <= 5; x++)
    {
        foo = getPixel(x, y);
        foo.setColor(Color.RED);
    }
}
```

Makes red box of

- width
- height

A. 10 5
B. 9 4
C. 5 10
D. 4 9
E. None of the above
public void foo(int x, int y) {
    Pixel foo;
    <<<LOOP HEADER 1>>>   {
        <<<LOOP HEADER 2>>>     {
            foo = getPixel(w,h);
            foo.setColor(Color.BLACK);
        }
    }
}

1) Solo: (1 min)
2) Discuss: (2 min)
3) Group: (30 sec)

A) for (int w = 0; w <= x; w++)
   for (int h = 0; h <= y; h++)
B) for (int w = 10; w < x+10; w++)
   for (int h = 20; h < y+20; h++)

C) for (int w = 0; w < y; w++)
   for (int h = 0; h < x; h++)
D) for(int w = 10; w <= x+10; w++)
   for(int h = 20; h <= y+20; h++)
From the book: Cropping A Picture (page 147-148) – we’ll change a bit

• Example of:
  – Working with both the calling object and a parameter object in a method
    • Extra information is passed to methods through parameters. The calling object is something like an extra parameter, named this
  – Doing something to a subset of the possible pixels in a picture
What part of Katie are we copying?  
(slight mod from the book)

public void copyKatiesXXX(Picture sourcePic)
{
    Pixel sPixel = null, tPixel = null;
    for (int sX = 40, tX = 100; sX < 110; sX++, tX++)
    {
        for (int sY = 350, tY = 100; sY < 400; sY++, tY++)
        {
            sPixel = sourcePic.getPixe(sX,sY);
            tPixel = this.getPixe(tX,tY);
            tPixel.setColor(sPixel.getColor());
        }
    }
}

A. Feet
B. Part of dress
C. Hands
D. Part of Couch
E. Face
REVIEW: Parameters (getting information into methods)

• It’s nice to have code that is “user controllable”…

• We have been hard-coding constants (40, 3, 100, for example) a lot, but we can write more flexible code using **PARAMETERS**

• This lets us write code to do things like “cropping and pasting into a blank canvas”, but letting the user specify what part of the source picture to crop, and where to place it in the canvas.
Underline the values you would change into parameters and write a new method header

```java
public void copyKatiesXXX( Picture sourcePic,
    int sxMin, int sxMax, int syMin, int syMax
)
{
    Pixel sPixel, tPixel = null;
    for (int sX = 40, tX = 100; sX < 110; sX++, tX++)
    {
        for (int sY = 350, tY = 100; sY < 400; sY++, tY++)
        {
            sPixel = sourcePic.getPixel(sX,sY);
            tPixel = this.getPixel(tX,tY);
            tPixel.setColor(sPixel.getColor);
        }
    }
}
```
TODO

• Reading for next class: 6.1-6.4

• Finish PSA5