

# CSE 8A Lecture 8

- Reading for next class: 6.1
- PSA4: DUE 11:59pm tonight (Collage and Picture Flip)
- PSA3 Interview, deadline Thursday 2/7 noon!
- Exams will promptly be started at beginning of class
  - Read your UCSD email the previous day for your new Exam seat assignment

**CLICKERS OUT!**

- |                             |
|-----------------------------|
| 1) Solo: (30 sec)           |
| 2) Discuss/Group:<br>(2min) |

# Mirroring Around Vertical Axis: Left to Right

- What are the parameter values we use to index leftPixel and rightPixel for the first three iterations of the inner loop? (assume picture has a height = 50 and width = 100)

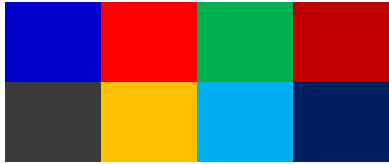
```
int mirrorPt = getWidth()/2;
Pixel leftP, rightP;
for (int y = 0; y < getHeight(); y++)
{
    for (int x = 0; x < mirrorPt; x++)
    {
        leftP = getPixel(x,y);
        rightP = getPixel(getWidth()-1-x,y);
        rightP.setColor(leftP.getColor());
    }
}
```

# How do you figure these kinds of questions out?

- Answer: Draw a diagram
  - imagine “beginning” and “answer”
  - Draw arrows to show how to get from beginning to answer
  - Then fill in numbers in order, write loops to create those numbers



# Mirroring Even Width versus Odd Width



```
int mirrorPt = getWidth() / 2;  
...  
for (int x = 0; x < mirrorPt; x++)
```

- 1) Solo: (30 sec)
- 2) Discuss/Group: (2 min)

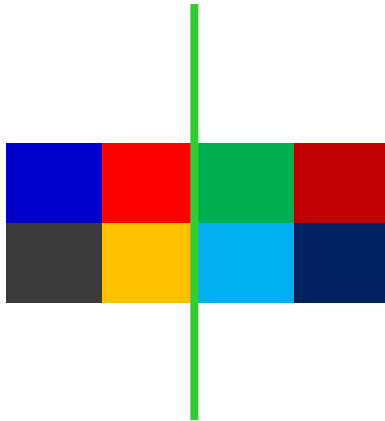
# Mirroring Odd-width Pictures

- What happens when this code attempts to mirror a Picture around the vertical axis when the Picture's width is odd (e.g. 101)?

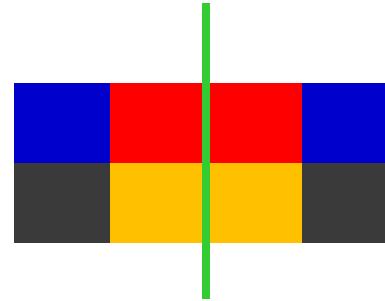
```
int mirrorPt = getWidth()/2;
Pixel leftP, rightP;
for (int y = 0; y < getHeight(); y++)
{
    for (int x = 0; x < mirrorPt; x++)
    {
        leftP = getPixel(x,y);
        rightP = getPixel(getWidth()-1-x,y);
        rightP.setColor(leftP.getColor());
    }
}
```

- A. It will work fine
- B. It will run, but it won't mirror correctly
- C. I won't run, there will be an index out of bounds exception
- D. It won't even compile if getWidth() is odd

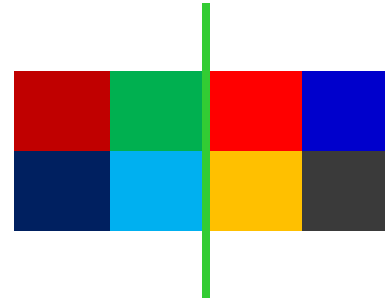
# Mirror versus “flip” (PSA4) (around vertical axis)



original



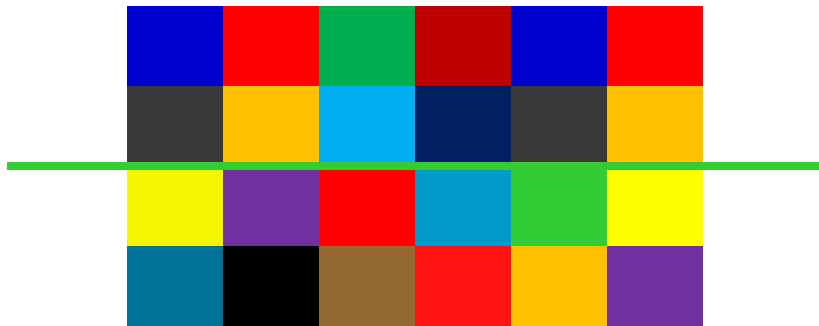
mirror



flip

- 1) Solo: (30 sec)  
2) Discuss/Group  
(1 min)

What are the first (x,y) coords for  
topP and bottomP  
to mirror around **horizontal** axis?



topP      bottomP

A. (0,0)      (0,3)

(0,1)      (0,2)

(1,0)      (1,3)

B. (0,0)      (0,3)

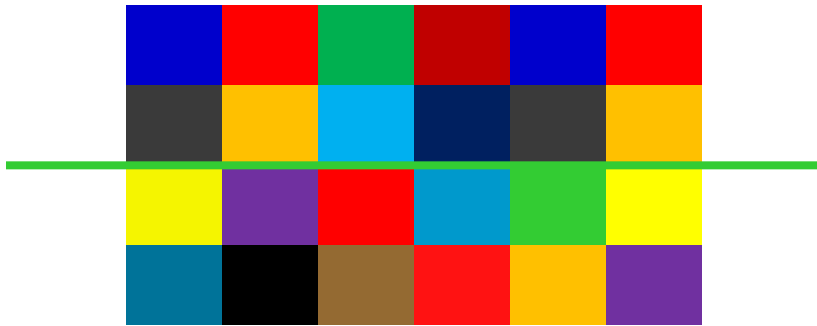
(1,0)      (1,3)

(2,0)      (2,3)

C. either A or B will work

D. none of the above

# Challenge: Complete the code that mirrors in the order specified by answer B



	topP	bottomP
B.	(0,0)	(0,3)
	(1,0)	(1,3)
	(2,0)	(2,3)

```
int height = getHeight();
```

```
int width = getWidth();
```

```
int mid = height/2;
```

```
Pixel topP, botP;
```

```
for (
```

```
) {
```

```
    for(
```

```
    ) {
```

```
        topP = getPixel(
```

```
        );
```

```
        botP = getPixel(
```

```
        );
```

```
        botP.setColor(topP.getColor());
```

```
    }
```

```
}
```



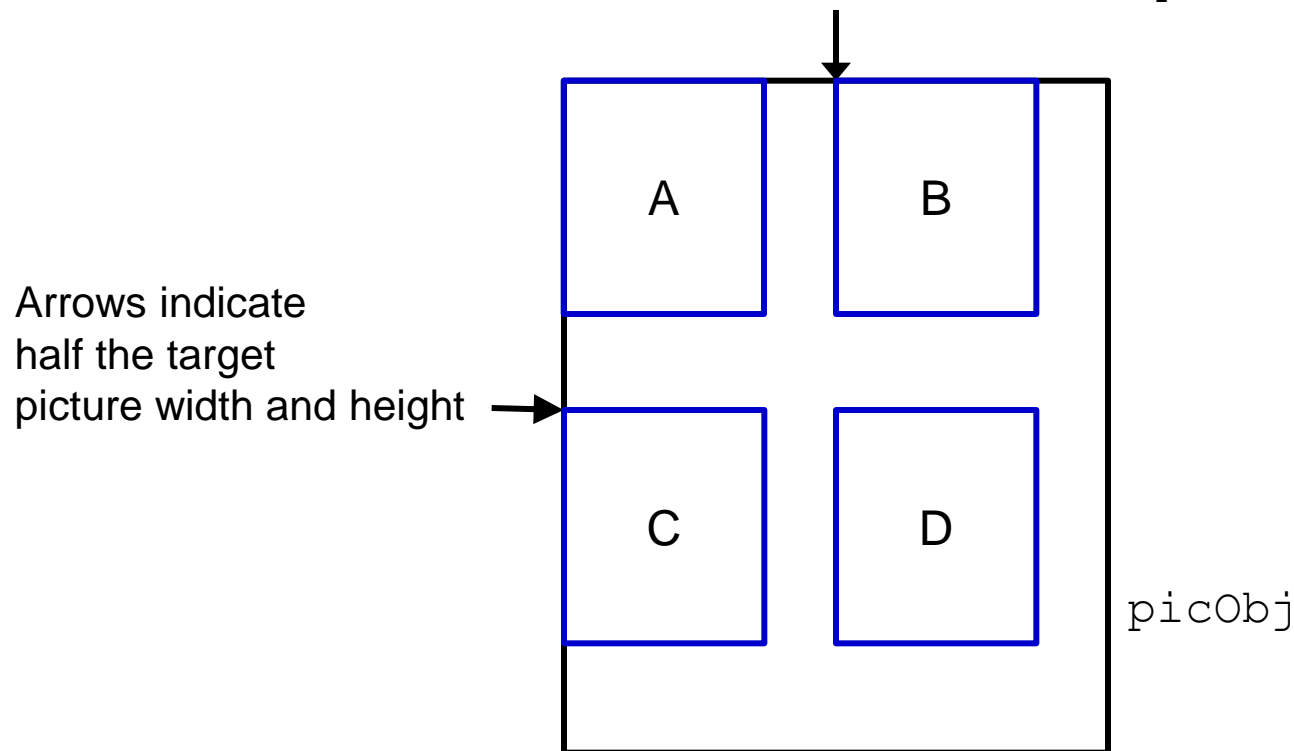
4. Imagine that you have a method `copyPictureTo`, whose header is below, defined in `Picture.java`. This method copies the source `Picture` (`sourcePic`) and places its top left corner at `(xStart, yStart)` of the `Picture` that called the `copyPictureTo` method.

```
public void copyPictureTo(Picture sourcePic, int xStart, int yStart)
```

Assume you have already created two `Picture` objects: `sourcePic` and `picObj`, and you make the following call:

```
picObj.copyPictureTo(sourcePic, picObj.getWidth()/2, 0);
```

Which position below best represents where the `sourcePic` will be located in `picObj` after the call? Assume that `sourcePic` is much smaller than `picObj`.



# Order of copying pixels

- When mirroring, we need to copy certain pixels to certain other pixels
- It doesn't matter what order we copy in, as long as when we are done, pixels have been copied correctly
- Two most common orders:
  - **Row major order:** copy all the pixels in one row, then go on to the next row
  - **Column major order:** copy all the pixels in one column, then go on to the next column

# Mirroring around horizontal axis

## column-major order

```
int height = getHeight();
int width = getWidth();
int mid = height/2;
Pixel topP, botP;
for(int x=0; x<width; x++)
{
    for(int y=0; y<mid; y++)
    {
        topP = getPixel(x,y);
        botP = getPixel(x,height-1-y);
        // copy one to the other...
    }
}
```

	topP	botP
A.	(0,0)	(0,3)
	(0,1)	(0,2)
	(1,0)	(1,3)
	...	...

## row-major order

```
int height = getHeight();
int width = getWidth();
int mid = height/2;
Pixel topP, botP;
for(int y=0; y<mid; y++)
{
    for(int x=0; x<width; x++)
    {
        topP = getPixel(x,y);
        botP = getPixel(x,height-1-y);
        // copy one to the other...
    }
}
```

	topP	botP
B.	(0,0)	(0,3)
	(1,0)	(1,3)
	(2,0)	(2,3)
	...	...

- 1) Solo: (1 min)
- 2) Discuss/Group: (2 min)

# 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--;
    }
}
```

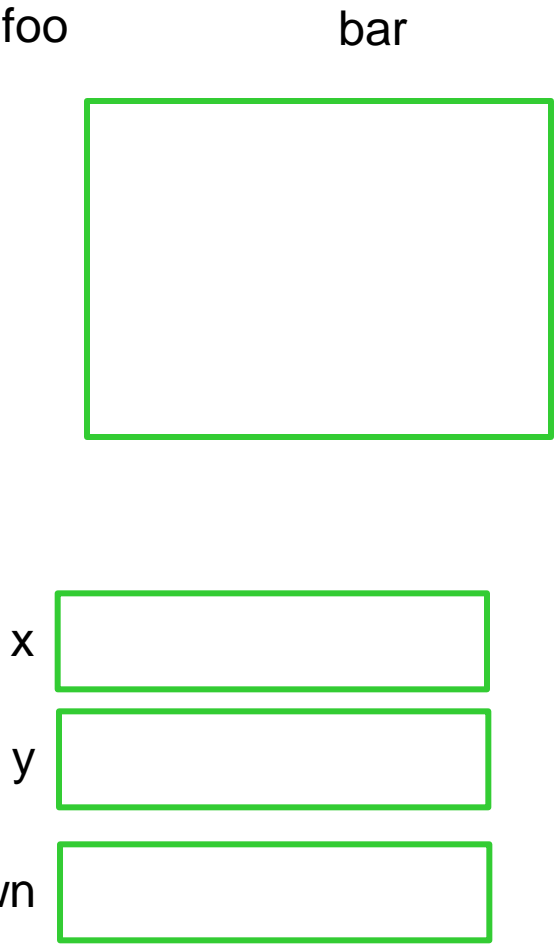
- A. Copies top half into bottom half not mirrored.
- B. Copies left half into right half not mirrored.
- C. Mirrors around vertical axis, left into right
- D. Mirrors around horizontal axis, top into bottom
- E. Some other bizarre transformation

- 1) Solo: (1 min)  
2) Discuss/Group:  
(2 min)

# 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--;  
    }  
}
```



By what variable name do we refer to collage inside makeC in Picture.java?

```
public class Lab4
{
    public static void main(String[] args)
    {
        Picture collage = new Picture("blank.jpg");
        Picture p = new Picture("bird1.jpg");
        Picture q = new Picture("bird2.jpg");
        collage.makeC(p,q);
    }
}
```

- A. collage
- B. callingObject
- C. Object
- D. Picture
- E. this

```
public class Picture
{
    public void makeC(Picture p1, Picture p2)
    {
        Pixel[] targetPixels = _____.getPixels();
        // ... more code
    }
}
```

# 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);`
- D. `Picture p =  
new Picture(aNum, bNum);`

- 1) Solo: (1 min)
- 2) Discuss/Group: (2 min)

# 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



- 1) Solo: (1 min)
- 2) Discuss/Group: (2 min)

# 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

- 1) Solo: (1 min)
- 2) Discuss/Group: (2 min)

# What are correct loops to make a black box of width x and height y?

```
public void foo(int x, int y) {  
    Pixel foo;  
    <<<<LOOP HEADER 1>>>> {  
        <<<<LOOP HEADER 2>>>> {  
            foo = getPixel(w,h);  
            foo.setColor(Color.BLACK);  
        }  
    }  
}
```

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++)

# TODO

- Reading for next class: 6.1
- PSA4: DUE 11:59pm tonight (Collage and Picture Flip)
- Don't forget your PSA3 interview!

