• Reading for next class: 6.4-6.5

• PSA5: Posted (due Monday 11:59pm)

• Today (random useful stuff):
  – class “static” methods
  – if else statements
Reading Quiz

CLICKERS OUT
1) What gets printed in the following:

```java
int x = 15, r = 10;
if( x < 15 )
{
    x = 100;
}
else
{
    r = r + 20;
}
System.out.println( x + r );
```

A. 115  
B. 45   
C. 125  
D. 145
2. What does the code above print?
A. Dogs can fly
B. Cats can drive
C. Non-Sensical nonsense
D. Nothing is printed
Does the order of the for loops matter?

```java
public void fillBottom( Color newColor )
{
    Pixel pix;

    for (int y = 0; y < this.getHeight(); y++)
    {
        for (int x = 0; x < this.getWidth(); x++)
        {
            <<<SELECT LINE OF CODE>>>>
            {
                pix = this.getPixel(x,y);
                pix.setColor(newColor);
            }
        }
    }
}
```

A. Yes, since we are changing the bottom half, we have to “fill in” across the rows in the inner loop
A. Yes, because we need to make sure the if statement is checking y not x
B. No, the if statement controls the assignment
How many times is the variable pix assigned a value?

A. 1
B. this.getWidth() times
C. this.getHeight() times
D. this.getHeight() * this.getWidth() times
E. this.getHeight()/2 * this.getWidth() times
A stripy filter

public void everyOtherColumn(Color newColor) {
    Pixel pix;

    for( int y = 0; y < this.getHeight(); y++ )
    {
        for( int x = 0; x < this.getWidth(); x = x + 2 )
        {
            pix = this.getPixel( x,y );
            pix.setColor( newColor );
        }
    }
}
How many iterations of the loop body are executed?

A. \((\text{getHeight}()-1) \times (\text{getWidth}()-1)\)

B. \((\text{getHeight}()-1) \times (\text{getWidth}()-1)/2\)

C. \(\text{getHeight}() \times \text{getWidth}()\)

D. \(\text{getHeight}() \times \text{getWidth}()/2\)

E. None of the above are always true

```java
public void everyOtherColumn(Color newColor) {
    Pixel pix;

    for (int y = 0; y < this.getHeight(); y++) {
        for (int x = 0; x < this.getWidth(); x = x + 2) {
            pix = this.getPixel(x,y);
            pix.setColor(newColor);
        }
    }
}
```
public void everyOtherColumn( Color newColor )
{
    Pixel pix;

    for (int y = 0; y < this.getHeight(); y++)
    {
        for (int x = 0; x < this.getWidth(); x++)
        {
            <<SELECT LINE OF CODE TO GO HERE>>
            pix = this.getPixel(x,y);
            pix.setColor(newColor);
        }
    }
}
If you can do it both ways, which is “better”?

• Efficiency
  – How much time (computer instructions) does it take?
  – How much space (computer memory) does it take?

• Software Design
  – How readable is it for a human?
  – How easy is it to modify the code if needed, without introducing new bugs?
Which is more efficient?

• We just looked at these two approaches for setting pixel color in even-indexed columns:
  – (A) Looping over only some (x,y) pixels
  – (B) Looping over all pixels, with if statement inside

• Which is more efficient?

• Why?
More complex control for color change

• What’s the most efficient way to reduce red eye?

A) Restricted loop bounds, only loop over x,y in range

B) Unrestricted loop bounds, use if statement to check for x,y in range

Pixels change if meet both a COLOR and COORDINATE criteria
public void removeRedEye( Color newColor,
        int startx, int endx, int starty, int endy )
{
    Pixel pix;
    for (int x = ; x < ; x++)
    {
        for (int y = ; y < ; y++)
        {
            pix = this.getPixel(x,y);
            if (pix.colorDistance(Color.red) < 167)
                pix.setColor(newColor);
        }
    }
}
Let’s do it the “inefficient” way

• Assume we loop over ALL pixels in picture…

• What if statement would we use to check for pixels in range?
public void removeRedEye(Color newColor, int startx, int endx,
int starty, int endy)
{
    Pixel pix;

    for( int x = 0; x < getWidth(); x++ )
        for( int y = 0; y < getHeight(); y++ )
            { 
                if( 
                    
                )
                { 
                    pix = this.getPixel(x,y);
                    if( pix.colorDistance(Color.red) < 167 )
                        pix.setColor(newColor);
                }
            }
}
Real-world example of if-else rules (among other things)

Eamonn Keogh, UCR

if (temp > 104.0)  
ALARM!  
Temp too high!
Real-world example of if-else rules (among other things)

Eamonn Keogh, UCR

TRILLIONS of data points (or more!)

How long would it take to loop through this data? How could you find patterns?
Real-world example of if-else rules (among other things)

Usually, the sensors only display the last few minutes of data and figures such as the minimum and maximum temperature for that day. In most cases, the rest of the data is discarded. This is in part due to legal and privacy issues, which the researchers believe can be solved. It’s also because computer scientists didn’t have the tools to mine the vast amounts of data produced in pediatric intensive care units. That changed after Keogh and a group of researchers recently developed a new technique, which allows for searching of datasets with more than one trillion objects. That’s a larger set than the combined size of all datasets in all data mining papers ever published.

http://ucrtoday.ucr.edu/9667
Keogh plans to use the archived data to develop algorithms that incorporate what he calls “if then rules” that can assist doctors. For example, if a heart beat looks like this, then a child may have difficulty breathing in five seconds.

http://ucrtoday.ucr.edu/9667
Unraveling the magic of `main`

Still a bit mysterious, for now

`public static void main(String[] args)`

Parameters (how are these passed in?)

Doesn’t return anything

Method name

Who “owns” the method…
Classes vs. Objects

In the code below, how many classes are there? How many objects? (And what does the memory model look like?)

```
Picture p = new Picture( "fish.jpg" );
Picture p2 = new Picture( p );
Picture p3 = p2;
```

A. 1 class, 2 objects
B. 1 class, 3 objects
C. 2 classes, 2 objects
D. 2 classes, 3 objects
E. 3 classes, 2 objects
In the code below, how many classes are there? How many objects? (And what does the memory model look like?)

```java
Picture p = new Picture("fish.jpg");
Picture p2 = new Picture(p);
Picture p3 = p2;
```
Static vs. nonstatic methods

**Static** methods can be called directly on the class (can also be called on an object)
**Non-static** methods must be called on an object

```java
public class Picture{
    ...

    public static Picture collage(Picture p1, Picture p2, Picture p3)

    public void filter1()
```

---

**Class - owned**

```
static
```

- Picture result = Picture.collision(p1, p2, p3);
- the Picture p1
- the Picture p2
- the Picture p3
- the Picture class
- the returned Picture

---

**Object - owned**

```
non static
```

- Picture p = new picture( "fish.jpg");
- p.filter1();
- the Picture p
- filter1
Summary of Concepts

• if else statements

• Efficiency

• class ("static") method vs ("instance") method
TODO

• Reading for next class: 6.4-6.5

• PSA5 due Monday 11:59pm