Iteration

- **While Loops**
  - **Syntax**
    ```
    initialize
    while (boolean_exp) {
      work_to_be_done
    update
    }
    ```
  - **Example**
    ```java
    int counter = 10;
    while (counter > 0) {
      System.out.println(counter);
      counter--;
    }
    System.out.println("Blast Off!");
    ```
  - **What is the output?**
  - **What if we exchange order of two statements in loop?**
Iteration

- For Loops
  - Syntax
    ```
    for (initialization; boolean_exp; update) {
        work_to_be_done
    }
    ```
  - Example
    ```
    for (int counter = 10; counter > 0; counter--) {
        System.out.println(counter);
    }
    System.out.println("Blast Off!");
    ```
  - What is the output?
  - When is update performed?
  - What is value of `counter` after loop?
Iteration

- **Do-While Loops**
  - Syntax
    
    ```
    initialize
    do
    {
      work_to_be_done
      update
    } while (boolean_exp);
    
    ```
    NOTE REQUIRED SEMICOLON!!
  
  - Example
    ```
    int counter = 10;
    do {
      System.out.println(counter);
      counter-- ;
    } while (counter > 0);
    System.out.println("Blast Off!");
    ```
Iteration

- Which Kind of Loop Do I Use?
  - While Loop
    - Don’t know how often it’s going be
    - Update can be anywhere in the loop body
  - For Loop
    - Know how often in advance
    - All information controlling loop together, in front
  - Do-While Loop
    - Least popular
    - Often used with data input

- What is the minimum number of times each of these loop?
  - while?
  - for?
  - do-while?
Iteration – Common Problems

- **Infinite Loops**
  - What causes them?
    - Always bad?

- **Semicolons**

- **Testing with != or ==**
  - Doubles
  - Increments other than 1

- **Scope of loop control variable (for loops)**

- **Off by one**
  - Starting at 0 vs 1
  - Using < vs <=
Iteration

- Can nest loops (and almost everything else)
  - Usually no special problems
  - Often used for matrices, tables, images . . .
  - Can hide by putting inner loop into helper method

- What is the output from the following?

```java
for (int k = 0; k < 8; k++) {
    for (int j = 0; j <= k % 4; j++) {
        System.out.print("*");
    }
    System.out.println();
}
```

- Trace!
Iteration

- While-loop controlled by Sentinel Value
  - Often loop continues until something is found or computed
  - Often involves I/O
    - Read input until something is encountered (sentinel) that means stop
  - When processing
    - Repeat until answer doesn’t improve any more

- Example: “indexOf(String s, char toFind)”

```java
int k = 0;
while (k < s.length() && s.charAt(k) != toFind) {
    k++;
}
if (k == s.length())
    return -1;
return k;
```
What is an image?

- Lots of pixels
- Each pixel represents a color

One color
RGB
(45, 10, 67)
Classwork today - Images

- Read in an image.
- Modify `transformColor` method
  - Given a Color, create and return a new Color
  - See API for Color class
  - “Magically” transformation is applied to all pixels in the image
  - Image is transformed to new image
  - Darken darkens the image
Example: Change red Color a little

```java
public Color transformColor (Color current) {
    int red = current.getRed();
    int blue = current.getBlue();
    int green = current.getGreen();
    return new Color(red + 50, green, blue);
}
```

CAREFUL!
What could possibly go wrong?