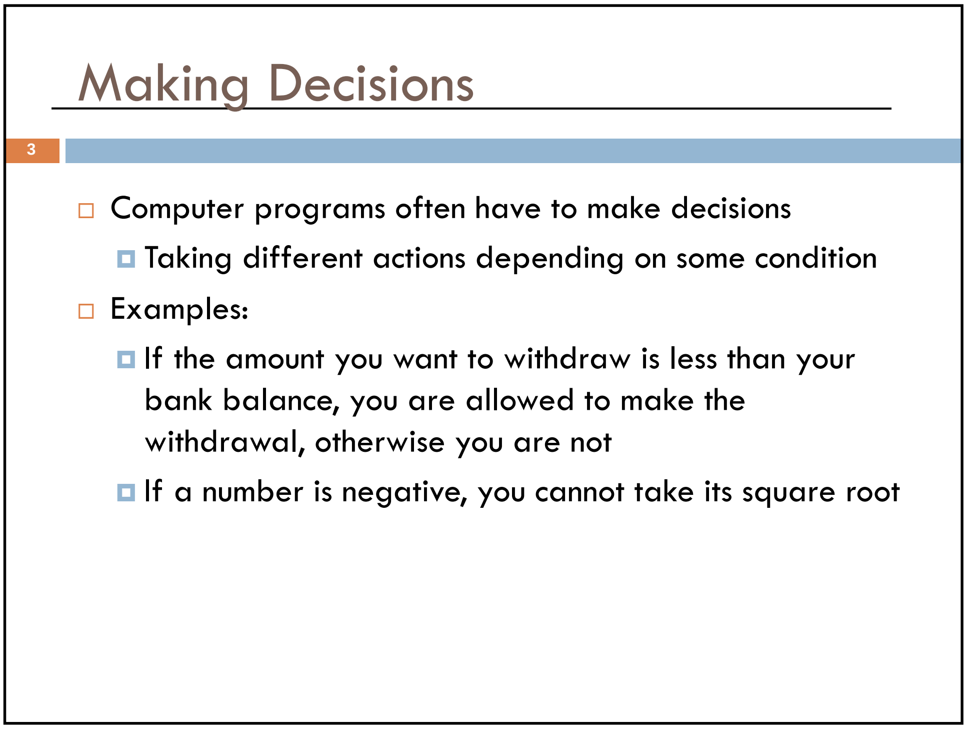
TOPIC 10

THE IF STATEMENT

Notes adapted from Introduction to Computing and Programming with Java: A Multimedia Approach by M. Guzdial and B. Ericson, and instructor materials prepared by B. Ericson.

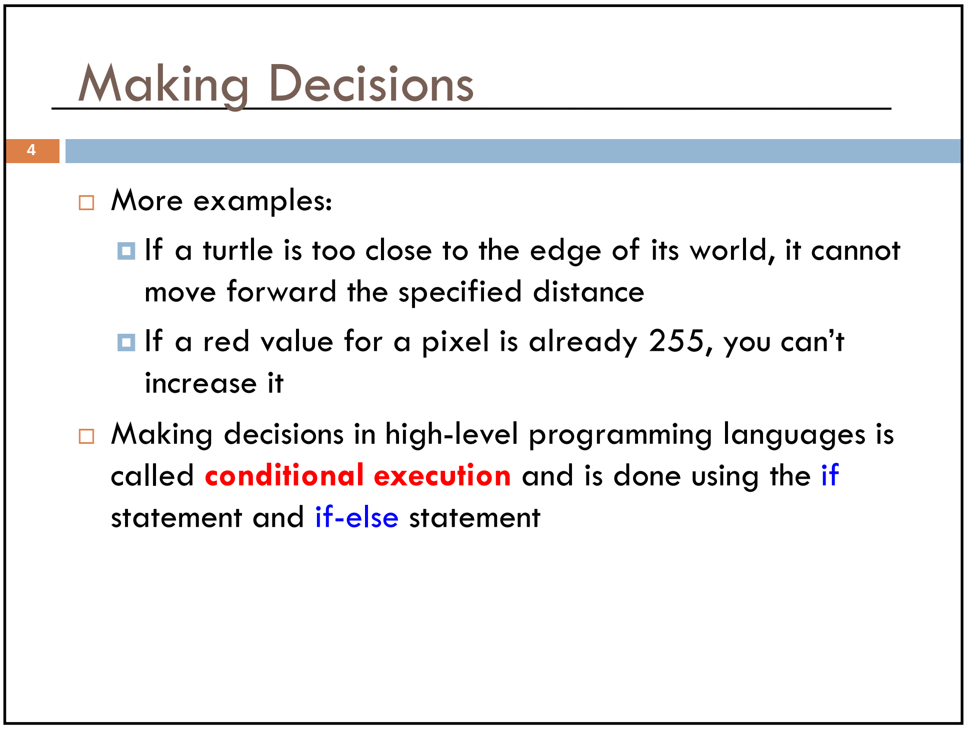
Outline

- How to use **conditionals**
- Picture manipulation using conditionals:
  - **Edge detection**
  - Sepia toning
  - Chromakey (Blue-screening)



Making Decisions

- Computer programs often have to make decisions
  - Taking different actions depending on some condition
- Examples:
  - If the amount you want to withdraw is less than your bank balance, you are allowed to make the withdrawal, otherwise you are not
  - If a number is negative, you cannot take its square root

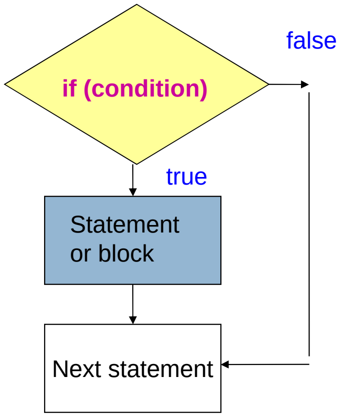


Making Decisions

- More examples:
  - If a turtle is too close to the edge of its world, it cannot move forward the specified distance
  - If a red value for a pixel is already 255, you can't increase it
- Making decisions in high-level programming languages is called **conditional execution** and is done using the if statement and if-else statement

If Statement

- A statement or block of statements is executed **only if** some condition is true
- If the condition is false, execution falls through to the next statement following the if statement



If Statement Syntax

- We may want a single statement to be executed if the condition is true:

```
if (condition)
   statement
```

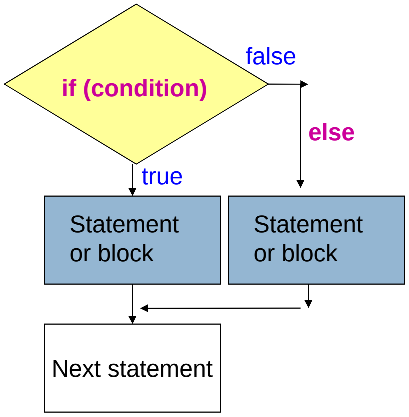
- We may want a block of statements to be executed if the condition is true:

```
if (condition)
{
   statement
    …
   statement
 }
```

- Safer to always use braces, even if a single statement
- For readability, make sure you line up the braces

If - else Statement

□ Using an if and else, we can do one thing if the condition is true
or a different thing if the condition is false



If - else Statement Syntax

□ If the condition is true, statement1 (or block of statements) is executed, otherwise statement2 (or block of statements) is executed

```
if (condition)
   statement1
else
   statement2
```

□ Example:

```
if (x < y)
   System.out.println("y is larger");
else
   System.out.println("x is larger");
```

Conditional Statement Example

- Bank withdrawal example:

```
// amount: the amount you want to withdraw
// balance: your account balance

if (amount <= balance)
{
    balance = balance – amount;
}
System.out.println("Balance is "  + balance);
```

- What happens if amount > balance?

Conditional Statement Example

- To let the user know that the withdrawal was not allowed

```
if (amount <= balance)
{
        balance = balance – amount;
        System.out.println("New balance is " + balance);
}
else
{
        System.out.println("Sorry, you are trying to withdraw
         $" + amount + " and your balance is only $" + balance);
}
```

Using Conditions in Pictures

- Choosing which pixels to manipulate in a picture
  - To remove red-eye in an image
  - To create sepia-toned images
  - To posterize images (reduce the number of color)
  - Do blue-screen effect
  - etc.

Simple Color Replacement

- Suppose we wanted to change all of the white in an image to a “bluer” white
- Algorithm:
  - For each pixel in the picture
    - If the pixel’s color is white
      - Decrease the red and green in this color

Simple Color Replacement Method

```
public void makeWhiteMoreBlue() {
    for (int x = 0; x < this.getWidth(); x++) {
        for (int y = 0; y < this.getHeight(); y++) {
           Pixel pixelObj = this.getPixel(x,y);
            int red = pixelObj.getRed();
            int green = pixelObj.getGreen();
            int blue = pixelObj.getBlue();
            if (red == 255 && green == 255 && blue == 255)
            {
               pixelObj.setRed(200);
               pixelObj.setGreen(200);
            }
        }
    }
}
```

A better method

- □ Allow more flexibility:
- □ Replace the color when all red, green and blue are above a threshold
- □ In this case, reduce red and green, and increase blue

Simple Color Replacement Method

```
public void makeWhiteMoreBlue() {
    for (int x = 0; x < this.getWidth(); x++) {
        for (int y = 0; y < this.getHeight(); y++) {
           Pixel pixelObj = this.getPixel(x,y);
           int red = pixelObj.getRed();
           int green = pixelObj.getGreen();
           int blue = pixelObj.getBlue();
           if (red >= 200 && green >= 200 && blue >= 200)
           {
               pixelObj.setRed(200);
               pixelObj.setGreen(200);
               pixelObj.setBlue(255);
           }
        }
    }
}
```

Multiple if statements

- Suppose we are doing different things based on a *set of ranges*, for example:

  0 <= x <= 5
  5 < x <= 10
  10 < x

```
if (0 <= x && x <= 5)
    statement or block

if (5 < x && x <= 10)
    statement or block

if (10 < x)
    statement or block
```

Using “else if” for > 2 Options

- Or we can use several if-else statements
- You don’t need to check if x > 5, since the first if block would have been executed if it was <= 5

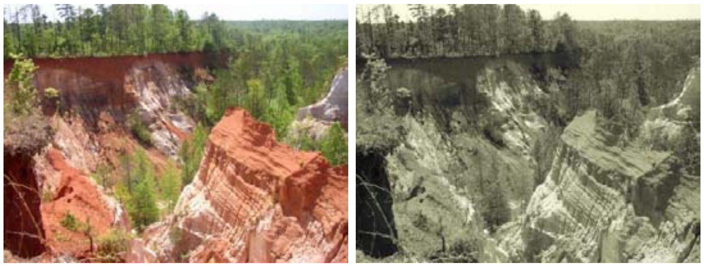
```
if (0 <= x && x <= 5)
    statement or block

else if (x <= 10)
    statement or block

else
    // what must x be?
    statement or block
```

Sepia-Toned Pictures

- They have a yellowish tint, used to make things look old

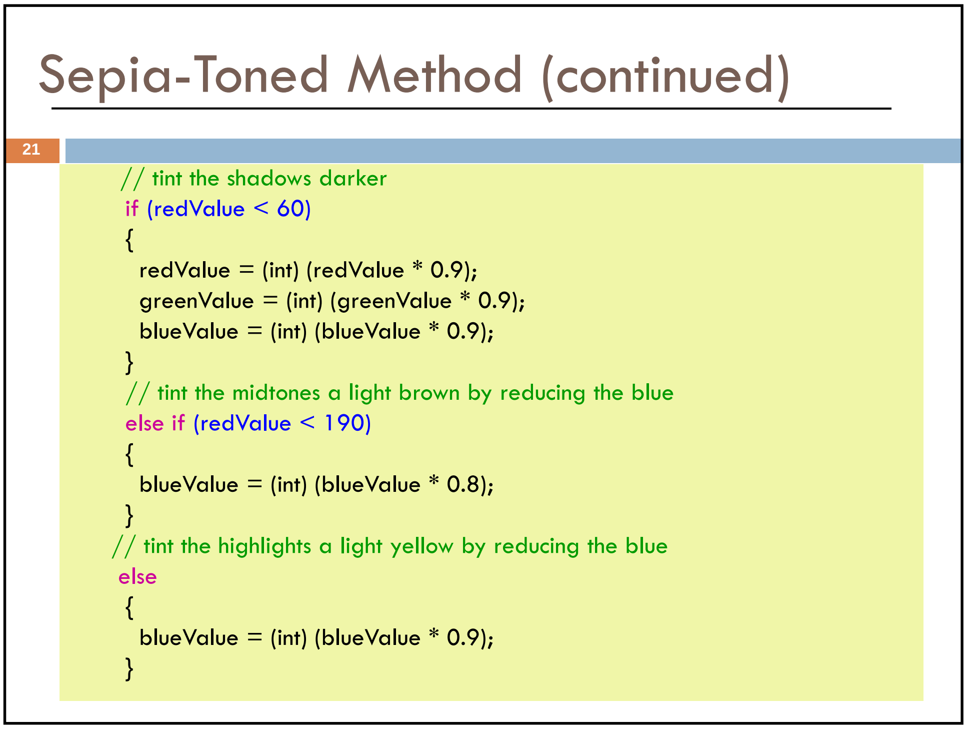


Sepia-Toned Algorithm

- First make the picture grayscale
- Change the darkest grays (shadows) to be even darker
- Make the middle grays a brown color
  - By reducing the blue
- Make the lightest grays (highlights) a bit yellow
  - By increasing red and green
  - Or decreasing blue (less than before though)
- We now have more than 2 possibilities, so we need to use multiple if statements

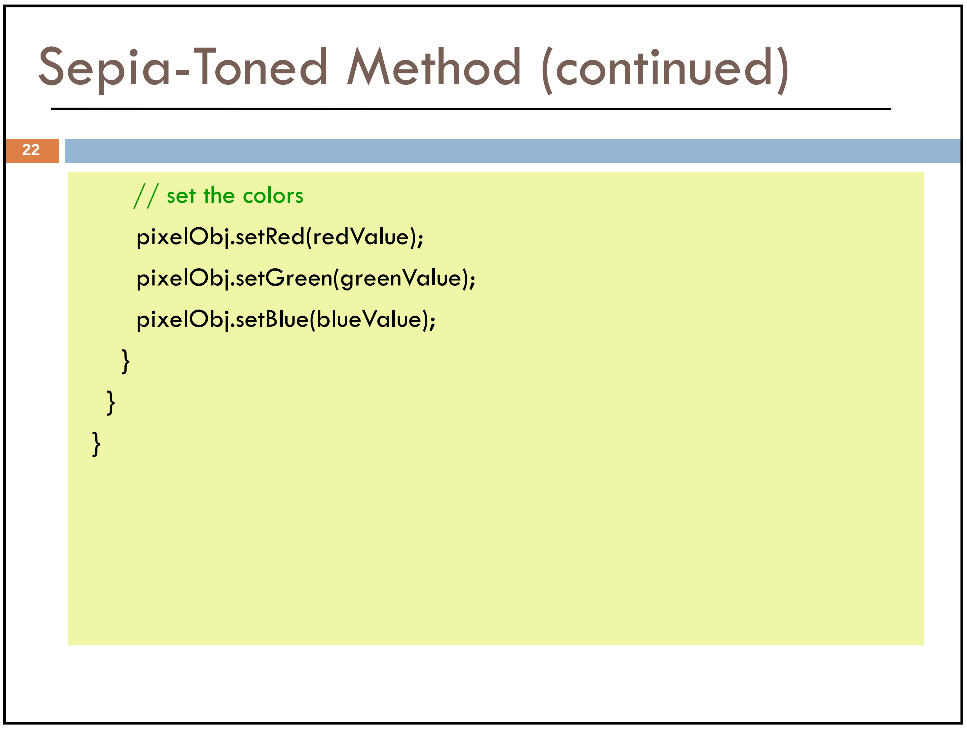
Sepia-Toned Method

```
public void sepiaTint()
 {
   // first change the current picture to grayscale
   this.grayscale();
    // loop through the pixels
   for (int x = 0; x < this.getWidth(); x++)
    {
      for (int y = 0; y < this.getHeight(); y++)
     {
       // get the current pixel and its color values
       Pixel pixelObj = this.getPixel(x,y);
       int redValue = pixelObj.getRed();
       int greenValue = pixelObj.getGreen();
       int blueValue = pixelObj.getBlue();
```



Sepia-Toned Method (continued)

```
    // tint the shadows darker
     if (redValue < 60)
     {
       redValue = (int) (redValue * 0.9);
       greenValue = (int) (greenValue * 0.9);
       blueValue = (int) (blueValue * 0.9);
     }
     // tint the midtones a light brown by reducing the blue
     else if (redValue < 190)
     {
       blueValue = (int) (blueValue * 0.8);
     }
   // tint the highlights a light yellow by reducing the blue
    else
     {
       blueValue = (int) (blueValue * 0.9);
     }
```



Sepia-Toned Method (continued)

```
       // set the colors
       pixelObj.setRed(redValue);
       pixelObj.setGreen(greenValue);
       pixelObj.setBlue(blueValue);
     }
   }
 }
```

Dangling Else Problem

- Consider this example:

```
if (x > 0)
        if (x < 100)
            System.out.println("one" );
else
        System.out.println("two");
```

  - Suppose x has the value −2. What do you think will be printed?

Dangling Else Problem

- The rule is that an else goes with the closest if
  - indentation makes no difference!
- Now what if we had wanted to group the else with the first if ? Use braces:

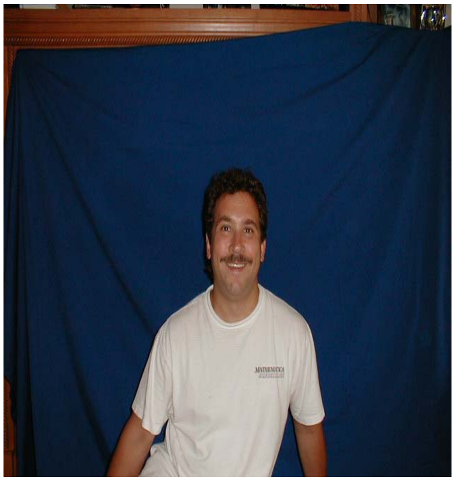
```
if (x > 0)
   {
     if (x < 100)
          System.out.println("one" );
   }
else
    System.out.println("two");
```

Blue Screen

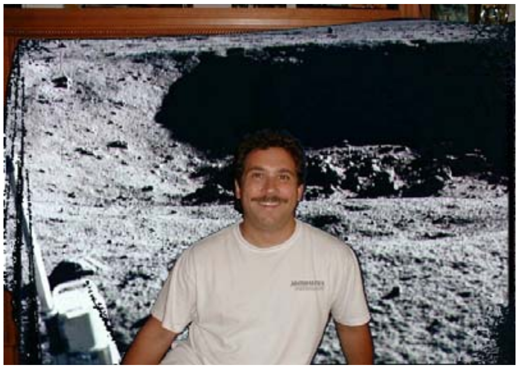
- The weather announcers on TV are not really standing in front of a changing weather map
  - They are in front of a solid colour background
    - blue or green (but usually blue – it’s better for skintones)
  - How do the weather map images appear behind them?
    - If a pixel’s color is blue, replace it with color of corresponding pixel from *new* background image
- Also used for special effects in TV and movies

Blue Screen

- Here just a blue sheet was used
- Professionally, you need an evenly lit, bright, pure blue background
- With nothing blue in the part of the image you want to keep



New Background



Algorithm

- Our method will be invoked on an image with a blue screen
  - The blue will be replaced as a background by another image
    - passed in as a parameter
- Algorithm:
  - Loop through all the pixels
  - If the pixel color is “blue”
    - Replace the pixel color with the color from the pixel at the same location in the new background picture

Method

```
public void blueScreen(Picture newBackground)
 {
   // loop through the columns
   for (int x=0; x<this.getWidth(); x++)
   {
     // loop through the rows
     for (int y=0; y<this.getHeight(); y++)
     {
       // get the current pixel
       Pixel currPixel = this.getPixel(x,y);
```

Method (continued)

```
        /* if the color of the pixel at this location is mostly blue
            (blue value greater than red and green combined),
            then use new background color at this pixel
         */
        if (currPixel.getRed() + currPixel.getGreen() <
                                                currPixel.getBlue())
        {
          Pixel newPixel = newBackground.getPixel(x,y);
          currPixel.setColor(newPixel.getColor());
        }
     }
   }
 }
```

Summary

- Conditionals
  - Nested Conditionals
  - Dangling Else
- Picture Algorithms that use conditionals