Making a Ball

- Adding a parametrized method “ball” to a class “Picture”
- Looping to get at many Pixels
- Our 2\textsuperscript{nd} if statement example: if a Pixel is in the ball then color it.
- Use of weighted average
- Method weAv that returns a (double) value.
- Method weAv is “static”
- Cast (int) required because truncating a double to an int LOSES INFORMATION.
Making a Ball

• Adding a parametrized method “ball” to a class “Picture”
  – pict.ball(xC,yC,radius) makes the computer work ON this Picture (referred to by the value of pict) to mess this Picture up by painting a ball onto it.
  – So, we say “ball” is an INSTANCE METHOD to mean ball works ON ONE (given) instance of the Picture class. That instance is “a Picture object”
Making a Ball

- Looping to get at many Pixels
  - The loop variables int x and int y hold coordinates, or positions or locations at which Pixel data is stored and changed.
Making a Ball

- Our 2\textsuperscript{nd} if statement example: if a Pixel is in the ball then color it.
  - The for loops restrict this testing and coloring if the Pixel is inside the ball to the smallest square region that contains the ball. This saves the computer time.
Making a Ball

• Use of weighted average
  − Makes the ball look round by coloring the middle white, the circumference a saturated red, and between, a smooth gradation of shades of red.

• Method weAv that returns a (double) value.
  − We wrote it ONCE AND FOR ALL, so if we need a weighted average calculated for some other reason, we do not have to bother coding the formula $N1*W1+N2*W2$ ever again.
Caller and Callee action for a method with a return value.

- Like before, the caller writes parameters on the calling card.
- Instead of doing something on say the caller's Picture, the caller computes some numbers and SHOUTS (communicates) a computed value back to the callee.
- Dramatized: Prof. calls weAv through a cell-phone, tells the callee 4.0, 0.5, 3.0, 0.5 The callee computes 3.5 and tells it to the Prof before she hangs up.
Making a Ball

- Method weAv is “static”
- Cast (int) required because truncating a double to an int LOSES INFORMATION.
• Method weAv that returns a (double) value.
  – We wrote it ONCE AND FOR ALL, so if we need a weighted average calculated for some other reason, we do not have to bother coding the formula \( N1 \times W1 + N2 \times W2 \) ever again.

• It's unnatural, distracting, misleading, and bad software expression to be forced to always have a Picture ON WHICH TO CALL weAv(N1,W1,N2,W2) So, we make it static method. (static is a Java keyword.)
  – That means weAv is always and only called ALL BY ITSELF, with 4 double numeric values.
  – “static” is a rather arbitrary word for the concept; it comes from Java's historical predecessors C and C++. 
The Math class

- Google it: Java API Math You can get accurate estimates of pi and e from Math.PI and Math.E
- We looked at the info about Math.sin().

- The Math class is a shopping bag holding a lot of methods that are called with numeric parameters, do popular mathematical calculations (for scientists, economists, sometimes business people).

- ALL the methods in Java's Math class are static
  - It makes no sense to force people to call them ON an object.
Instance methods

- Almost all the methods we covered: forward() (of Turtle), show( ) (of Picture and of Turtle), getWidth(), getHeight() and getPixel (of Picture), setColor() (of Pixel) are instance methods

They are ONLY called ON some object (like the Turtle, Picture, or Pixel) on which calling the method makes sense.