Overview
The purpose of class 4 is to look at types of loops and to get people to think in terms of loops. That means to think of a task as a repetition. Why use repetition, though?

To simplify a solution and also to build a framework that can grow.

We look at visual patterns for the simplest pattern that can be repeated. Sometimes the repetition requires variation. But that is for next time.

Principles of Programming
Larry Wall says the driving forces for good programming are laziness, impatience, and hubris. Ask people what each means and why it is good for programming. See Larry’s explanations for the answers.

Let’s keep those in mind as we look at some solutions

Two Versions of the 'find minimum card’ algorithm. One is 9 lines of repeated code, the other is two lines: a repeat and a conditional.

Let’s look at the anatomy and terminology of these two grammatical constructs. A loop has a control part and a body. A conditional statement has a boolean expression and an action.

But, computers are not the only place we see repetition with minor variations from iteration to iteration.

99 bottles of beer is an example we can recode using a loop.

Junk mail (Dear Tim, you may be a winner... ; Dear Anita, you may be a winner...)

Old MacDonald loops with minor variations. We look at two kinds of loops: numbers and lists. What part is the same and what part varies? That is the key to addressing the laziness principle. I don’t want to type 99 lines of song; let’s just put in a repeat and tell the user how to generate each line. Both methods (in-line, loop) work, but one is less work and more elegant (hubris).

Show the Old MacDonald video. Discuss sequence and well-defined. Be sure to focus on ‘sequence’ because it affects the way loops test the condition in sequence as opposed to as a rule.

Graphics and Loops and Variables
Now we can use these ideas to tell a computer how to build images.

How do humans draw pictures?
   Pick a color
   put the pen on the paper
   move the pen

Scratch works the same way. Here is a method for drawing a square. Let’s try it. What about the window picture? Try this at your desk.

Lots of solutions, but which one shows laziness? The simplest one. But the key is to find the repeated pattern. What is the simplest repeated pattern that will do it?

Then build flowers by changing the angle by a smaller amount. Then change colors by using 'change color
by [n]’ (this leads to the changing of variables later on.

**The Row of Boxes**

Let students work for 3-4 minutes. The discuss solutions. If the thing hits the wall, how to stop it. Some will say repeat until <touching edge> but that has problems.

Instead, repeat until (x position) > 240. But depending on width that will not work if the box straddles the edge. Which introduces the speeding limit vs the radar gun.

What if we want smaller boxes. We need to change two move commands. But we can simplify that with a variable.

**Summary**

Types of loops: forever, repeat N, repeat until <> . What are examples of each type in `real life`. Shampoo has the forever loop. People think that’s funny. Loops allow you to simplify work by repeating common patterns. But you have to ‘see’ the patterns to be able to use loops well. That is a creative part of problem solving -- seeing the problem as a pattern made of smaller patterns. Like fractals.

On student mentioned Stoppard’s Arcadia and the invention of fractals in the 19’th century.