CSC 108
Introduction to Computer Programming

Lecture 10

Repetition

University of Mount Union

The most important concept we'll study this semester

The key to the power of computer programming is to recognize repetitive patterns and GENERALIZE those patterns

Example: print the numbers from 1 to 10 on separate lines in the Console

The only way we could do this now would be to have 10 println statements

The Brute Force Solution

```
println( 1 );
println( 2 );
println( 3 );
println( 4 );
println( 5 );
println( 6 );
println( 7 );
println( 8 );
println( 9 );
println( 10 );
```

The Brute Force Solution

But what if we want all the values from 1 to 1000?

Brute Force solutions are correct, but they involve **way** too much work

Instead, what repetitive pattern can we notice?

We want to do the same thing to each of the values starting with 1 and going up to 10

- print the current value in the console

The key to understanding repetition in programming

writing the operation to be performed IN GENERAL
(that means using a variable, not a specific number)

The operation to be performed is

```
println( x );
```

We want to do that for all the values from 1 to 10

How can a variable be used in this situation

```
int x = 1;
println( x );
x = x + 1;
println( x );
x = x + 1;
println( x );
x = x + 1;
println( x );
x = x + 1;
println( x );
// continued over here
```

```
x = x + 1;
println( x );
x = x + 1;
println( x );
x = x + 1;
println( x );
x = x + 1;
println( x );
x = x + 1;
println( x );
x = x + 1;
```

What operations are being repeated?

Done once at the beginning:

```
int x = 1;
```

Done ten times in a row:

```
 println( x );
 x = x + 1;
```

The **for** Statement in Processing

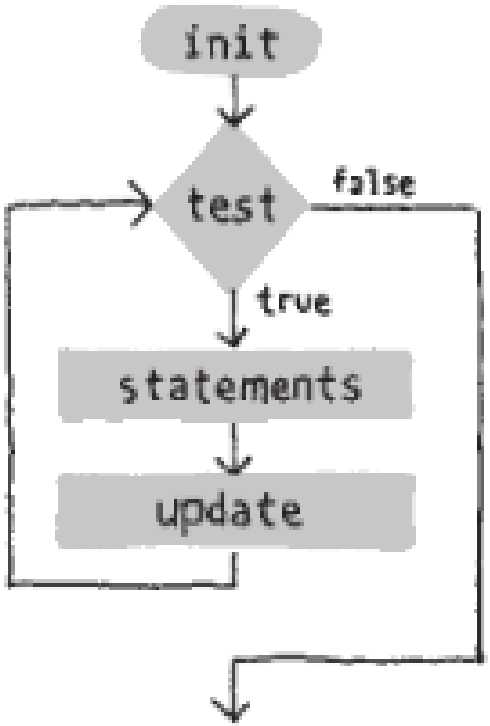
purpose: repeatedly execute a block of code a specific number of times

```
for ( int x = 1; x <= 10; x = x + 1 ) {
    println( x );
}
```

What if we want all the numbers from 1 to 1000?

Much easier this way!

The **for** Statement in Processing




```
for (init; test; update) {
  statements
}
```

Where's the repetition, Waldo?

Consider this example – what does it do?

```
size(500, 500);
rect( 20,  20,  30,  30 );
rect( 20,  60,  60,  30 );
rect( 20, 100,  90,  30 );
rect( 20, 140, 120,  30 );
rect( 20, 180, 150,  30 );
rect( 20, 220, 180,  30 );
rect( 20, 260, 210,  30 );
rect( 20, 300, 240,  30 );
```

What is changing from line to line, and what is remaining the same?

Where's the repetition, Waldo?

Everything is the same except the vertical position of each rectangle, and the width of each rectangle.

```
size(500, 500);
rect( 20,  20,  30,  30 );
rect( 20,  60,  60,  30 );
rect( 20, 100,  90,  30 );
rect( 20, 140, 120,  30 );
rect( 20, 180, 150,  30 );
rect( 20, 220, 180,  30 );
rect( 20, 260, 210,  30 );
rect( 20, 300, 240,  30 );
```

How can we generalize this idea and use repetition?

Where's the repetition, Waldo?

Write the statement ONCE, with a variable for what changes, and constants for everything else

```
rect( 20,  y,  width,  30 );
```

y should start at 20, width should start at 30

draw the rectangle, then increase y by 40 and increase width by 30

repeat this once for each rectangle to be drawn (in this case, that means 8 times)

Here's the repetition

```
size(500, 500);
int y = 20;
int width = 30;
for (int num = 1; num <= 8; num = num + 1 ) {
    // draw a rect
    rect( 20,  y,  width,  30 );
    // update y and width to the next value
    y = y + 40;
    width = width + 30;
} // end for
```

What if we want to draw 5 rectangles, or 12?

- just change the number of times the loop is executed!

CSC 108
Introduction to Computer Programming

Lecture 10

Repetition

University of Mount Union