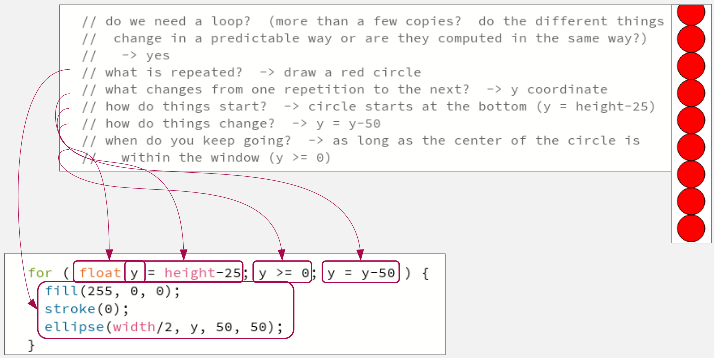
Loop Questions

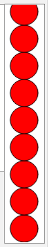
- What is repeated? → loop body
- What changes from one repetition to the next?
  → loop variables(s)
- How do things start?
  → initialization of loop variables
- How do things change?
  → update of loop variables
- When do you keep going? → loop condition
  - may be easier to think “when do you stop?” and then figure out the opposite
  - two patterns
    - repeat as long as / until – condition involves one or more loop variables
    - repeat *n* times (counting loop)

Repeat-As-Long-As Example



```
// do we need a loop?  (more than a few copies?  do the different things
//  change in a predictable way or are they computed in the same way?)
//   -> yes
// what is repeated?  -> draw a red circle
// what changes from one repetition to the next?  -> y coordinate
// how do things start?  -> circle starts at the bottom (y = height-25)
// how do things change?  -> y = y-50
// when do you keep going?  -> as long as the center of the circle is
//    within the window (y >= 0)
```

```
for ( float y = height-25; y >= 0; y = y-50 ) {
  fill(255, 0, 0);
  stroke(0);
  ellipse(width/2, y, 50, 50);
}
```



- for loop

```
for ( declare and initialize loop variables ;
      loop condition ; update loop variables ) {
   loop body
}
```

Counting Loops – Repeat *n* Times

- a *counting loop* is when the “keep going” condition involves a number of repetitions – repeat *n* times
  - introduce a counter variable to keep track of the number of repetitions completed so far

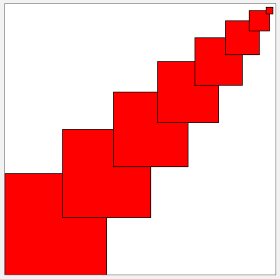
if the loop variables are all ints

```
for ( int count = 0,
      declare and initialize loop variables ;
      count < number of repetitions ;
      count = count+1, update loop variables ) {
   loop body
}
```

if the loop variables are not all ints

```
{
   declare and initialize loop variables
   for ( int count = 0 ;
         count < number of repetitions ;
         count = count+1, update loop variables ) {
     loop body
   }
}
```

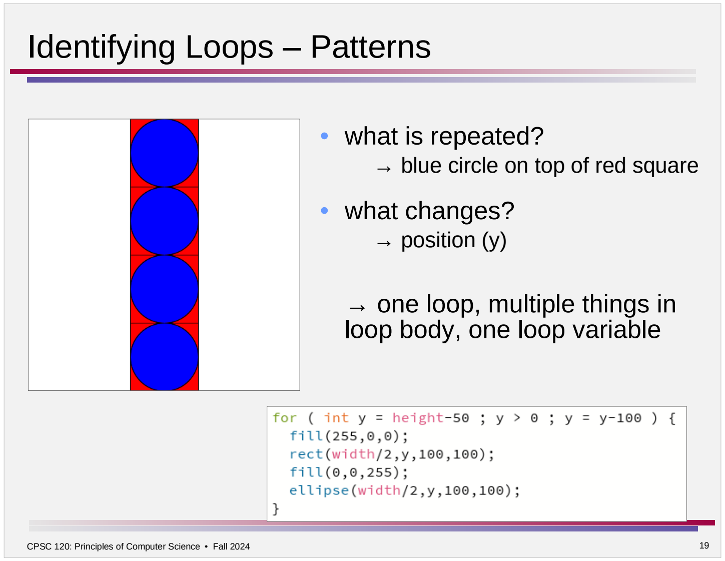
Identifying Loops – Patterns



- what is repeated?
  → red square
- what changes?
  → position (x, y) and size

→ one loop, one thing in loop body, multiple loop variables

```
for ( int x = 75, y = height-75, dim = 150 ;
      dim > 0 ;
      x = x+dim/2, y = y-dim/2, dim = dim-20 ) {
    rect(x,y,dim,dim);
}
```

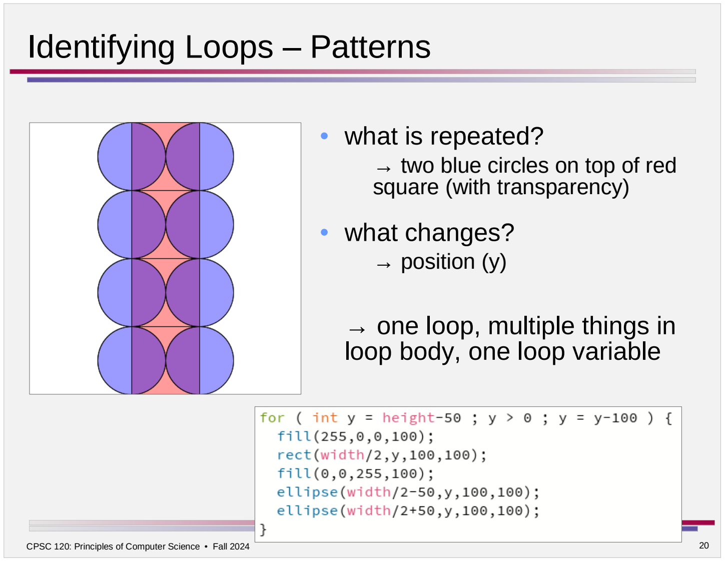


Identifying Loops – Patterns

- what is repeated?
  - → blue circle on top of red square
- what changes?
  - → position (y)

→ one loop, multiple things in loop body, one loop variable

```
for ( int y = height-50 ; y > 0 ; y = y-100 ) {
  fill(255,0,0);
  rect(width/2,y,100,100);
  fill(0,0,255);
  ellipse(width/2,y,100,100);
}
```

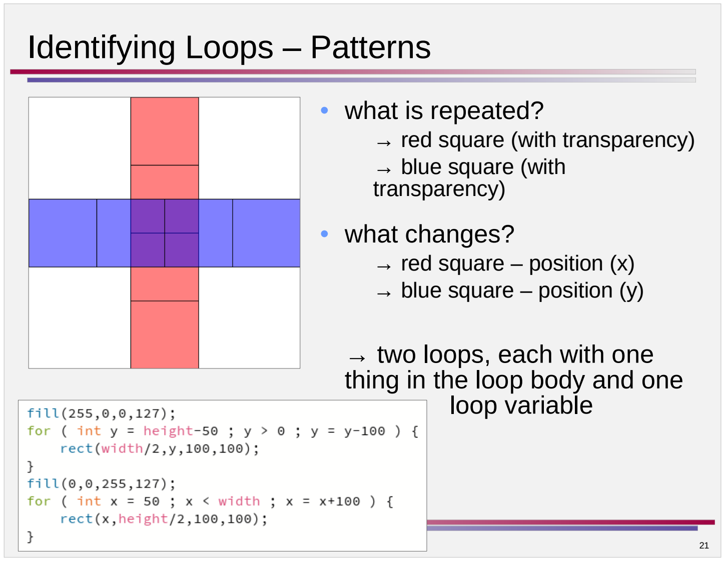


Identifying Loops – Patterns

- what is repeated?
  - → two blue circles on top of red square (with transparency)
- what changes?
  - → position (y)

→ one loop, multiple things in loop body, one loop variable

```
for ( int y = height-50 ; y > 0 ; y = y-100 ) {
  fill(255,0,0,100);
  rect(width/2,y,100,100);
  fill(0,0,255,100);
  ellipse(width/2-50,y,100,100);
  ellipse(width/2+50,y,100,100);
}
```

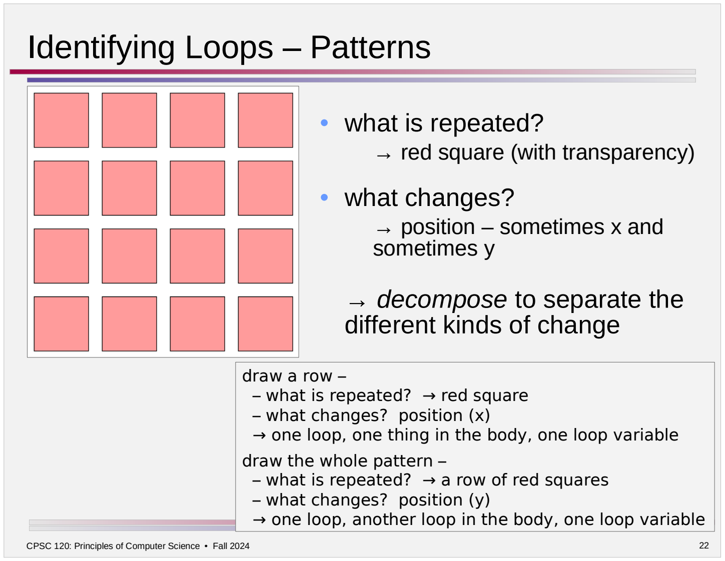


Identifying Loops – Patterns

- what is repeated?
  - → red square (with transparency)
  - → blue square (with transparency)
- what changes?
  - → red square – position (x)
  - → blue square – position (y)

→ two loops, each with one thing in the loop body and one loop variable

```
fill(255,0,0,127);
for ( int y = height-50 ; y > 0 ; y = y-100 ) {
    rect(width/2,y,100,100);
}
fill(0,0,255,127);
for ( int x = 50 ; x < width ; x = x+100 ) {
    rect(x,height/2,100,100);
}
```



Identifying Loops – Patterns

- what is repeated?
  - → red square (with transparency)
- what changes?
  - → position – sometimes x and sometimes y

→ *decompose* to separate the different kinds of change

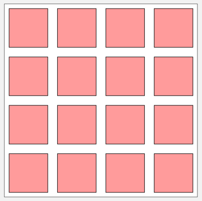
draw a row –
- what is repeated? → red square
- what changes? position (x)
- → one loop, one thing in the body, one loop variable

draw the whole pattern –
- what is repeated? → a row of red squares
- what changes? position (y)
- → one loop, another loop in the body, one loop variable

Identifying Loops – Patterns

draw a row –
- what is repeated? → red square
- what changes? position (x)
- → one loop, one thing in the body, one loop variable

draw the whole pattern –
- what is repeated? → a row of red squares
- what changes? position (y)
- → one loop, another loop in the body, one loop variable

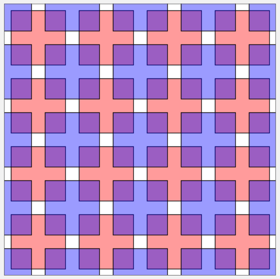


```
fill(255, 0, 0, 100);
// change the y coordinate as we move from
//  one row to the next
for ( int y = height-50; y > 0; y = y-100 ) {
  // draw one row

  // change the x coordinate as we move from
  //  one column to the next within one row
  for ( int x = 50; x < width; x = x+100 ) {
    // draw the rectangle!
    rect(x, y, 80, 80);
  }
}
```

draw the whole pattern

draw a row

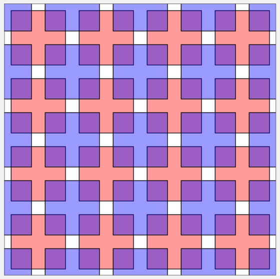
Identifying Loops – Patterns



- what is repeated?
  - → red square (with transparency)
  - → blue square (with transparency)
- what changes?
  - → position – sometimes x and sometimes y

→ two loops (red square, blue square), each a nested loop

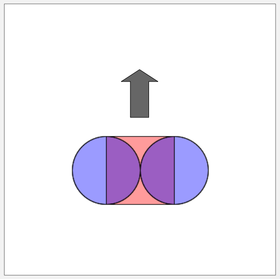
Identifying Loops – Patterns



→ two loops (red square, blue square), each a nested loop

```
// red squares
fill(255,0,0,100);
for ( int x = 50 ; x < width ; x = x+100 ) {
  for ( int y = height-50 ; y > 0 ; y = y-100 ) {
    rect(x,y,80,80);
  }
}

// blue squares
fill(0,0,255,100);
for ( int x = 0 ; x <= width ; x = x+100 ) {
  for ( int y = height ; y >= 0 ; y = y-100 ) {
    rect(x,y,80,80);
  }
}
```

Identifying Loops – Patterns

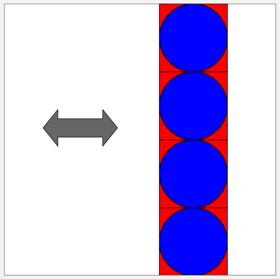


- what is repeated?
  - → nothing

→ animation only, no loops
- one animation variable (y)

```
float y;

void setup () {
  size(400,400);
  y = height-50;
}

void draw () {
  background(255);
  rectMode(CENTER);

  fill(255,0,0,100);
  rect(width/2,y,100,100);
  fill(0,0,255,100);
  ellipse(width/2-50,y,100,100);
  ellipse(width/2+50,y,100,100);

  y = y-.3;
}
```

Identifying Loops – Patterns



- what is repeated?
  - → blue circle on top of red square
- what changes?
  - → position (y)

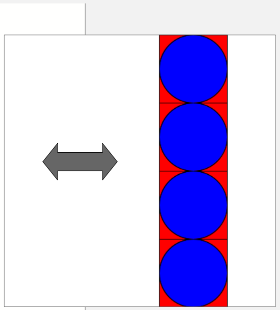
→ animation + one loop, multiple things in the loop body, one loop variable

- one animation variable (x) + state machine pattern

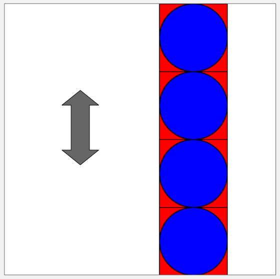
```
float x;          // x coordinate of the center of the column
boolean goleft;   // if true, move to the left, otherwise move right

void setup () {
  size(400, 400);
  x = width/2;
  goleft = false;
}

void draw () {
  background(255);
  rectMode(CENTER);

  // draw the column of shapes
  for ( int y = height-50; y > 0; y = y-100 ) {
    fill(255, 0, 0);
    rect(x, y, 100, 100);
    fill(0, 0, 255);
    ellipse(x, y, 100, 100);
  }

  // update the animation variable for next time
  if ( goleft ) {
    x = x-1;
  } else {
    x = x+1;
  }

  // update the state variable for next time
  if ( x >= width-50 ) {
    goleft = true;
  } else if ( x <= 50 ) {
    goleft = false;
  }
}
```

→ animation + one loop, multiple things in the loop body, one loop variable

- one animation variable (x) + state machine pattern



Identifying Loops – Patterns



- what is repeated?
  - → blue circle on top of red square
- what changes?
  - → position (y)

→ animation + one loop, multiple things in the loop body, one loop variable

- one animation variable (y) + bouncing

```
float starty;      // the y coordinate of the center of the bottom circle/square combo
float yspeed;      // the amount the y coordinate changes in one step

void setup () {
  size(400,400);
  starty = height/2;
  yspeed = 1;
}

void draw () {
  background(255);
  rectMode(CENTER);

  // repeated: the circle/square combo
  // changing: the y coordinate of the center of the circle/square combo
  // loop variable: y
  for ( float y = starty ; y > -50 ; y = y-100 ) {
    fill(255,0,0);
    rect(width/2,y,100,100);
    fill(0,0,255);
    ellipse(width/2,y,100,100);
  }

  starty = starty+yspeed;
  yspeed = yspeed+.1;
  if ( starty >= height-50 ) {
    yspeed = -.9*yspeed;
    starty = height-50;
  }
}
```

→ animation + one loop, multiple things in the loop body, one loop variable

- one animation variable (y) + bouncing