The if statement

```java
if (BooleanExpression) {
    Statement
}
else {
    Statement
}
```

The if-else statement

```java
if (BooleanExpression) {
    Statement1
} else {
    Statement2
}
```

What goes in the blank so that the ball moves back and forth, reversing direction whenever it reaches the edge?

A. Leave it blanks (as is).
B. if (ballX > ballDia/2)
C. if (ballX < ballDia/2)
D. (ballX > ballDia/2)
E. (ballX < ballDia/2)
Semicolons and the if statement

```java
if (carX < 0)
    carX = width;
// draw car
...

if (carX < 0) {
    carX = width;
    carY = carY + 10; // move down each time it wraps around
}
```

Semicolons and common error

```java
void draw() {
    if (carX < 0);
    carX = width;
    // draw car
    ...
}
```

Logical Operators

- Operators that take boolean values as operands.
- `x && y` - true if `x` AND `y` are both true
- `x || y` - true if either `x` OR `y` are true, or both
- `!x` - true if `x` is false - read NOT `x`

```java
void setup() {
    size(200, 200);
    rectMode(CORNERS);
}
int boxLeft = 50, boxRight = 150,
    boxTop = 50, boxBottom = 150;
void draw() {
    if (mouseX > boxLeft && mouseX < boxRight &&
        mouseY > boxTop && mouseY < boxBottom )
        fill(255,0,0);
    else{
        fill(0,255,0);
    }
    rect(boxLeft, boxTop, boxRight, boxBottom);
}
```

Which expression completes this program so that it shows a red circle when the mouse is inside of the circle and a green circle when the mouse is outside of the circle?

A. `dist(mouseX, mouseY, circleX, circleY) <= dia/2`
B. `dist(mouseX, mouseY, circleX, circleY) >= dia/2`
C. `dist(mouseX, mouseY, circleX, circleY) < dia`
D. `dist(mouseX, mouseY, circleX, circleY) > dia`
E. `abs(mouseX-circleX) < dia/2 && abs(mouseY-circleY) < dia/2`

Bouncing Ball

- `pos_{t+1} = pos_t + velocity`
- `velocity_{t+1} = velocity_t + acceleration`
- gravity provides constant acceleration downward
float velocity = 2;
float yPos = 0;
int ballRadius = 10;

void draw() {
  background(255);
  // if hit the ground reverse the velocity
  if (yPos > height-ballRadius) {
    velocity = -velocity;
  }
  // adjust position based on velocity
  yPos = yPos + velocity;
  // draw the ball
  ellipse(width/2, yPos, ballRadius*2, ballRadius*2);
}

Recap
• boolean valued expressions using relational operators: <, >, <=, >=, ==, !=
• boolean operators &&, ||, !
• if (booleanExpression) {
  sequence of statements
} else {
  sequence of statements
}

For loops (Repetition)
• Repeating commands is powerful:
– Lightbot 2.0 used recursion, a function calling itself
– Symbolic Lightbot prefixed a number, 2:Step
• Processing uses a for loop:

```java
for (int i=0; i < 16; i++) {
  ellipse(100+25*i, 100, 15, 15);
}
```

Repetition, the Picture
• A for loop has several parts, all required ...

```java
for (int i=0; i < 16; i++) {
  ellipse(100+25*i, 100, 15, 15);
}
```

The result of this statement is 16 copies of the stuff to be repeated. 16 Pacman pills
Repetition, Another Picture

- Or how about a bullseye?

- Note the loop variable must be declared ... but could do it in loop itself like we did for pacman pills:

- for (int i = 0; ...