Programming Fundamentals 1

Produced by

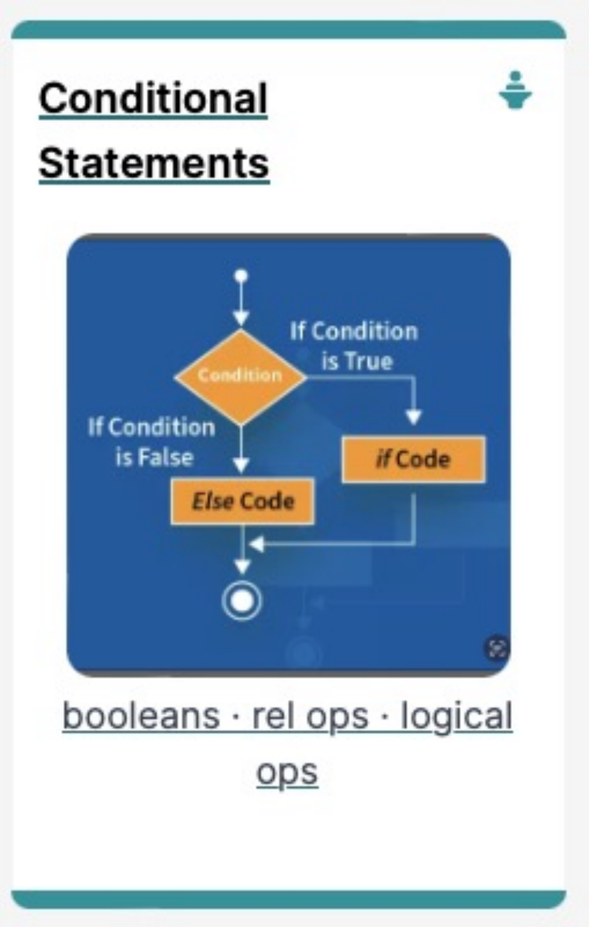
Mr. Dave Drohan (david.drohan@setu.ie)
Dr. Siobhán Drohan
Ms. Mairead Meagher

Department of Computing & Mathematics
South East Technological University
Waterford, Ireland

setu.ie

Introduction to Processing

Conditional Statements and Boolean Expressions



Conditional Statements

booleans · rel ops · logical ops

Agenda

❑ Conditional Statements

❑ Boolean Conditions & Relational Operators

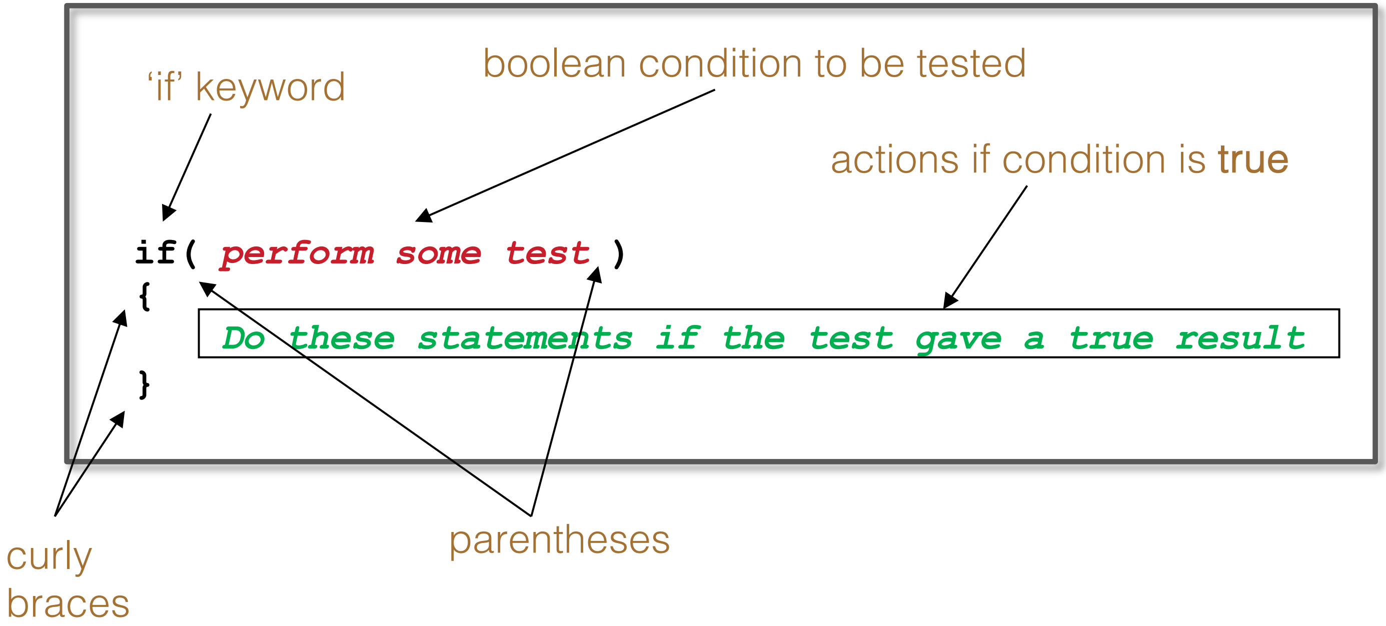
❑ Logical Operators

Conditional Statements

Conditional Statement Syntax (1)

```
if( perform some test )
{
    Do these statements if the test gave a true result
}
```

Conditional Statement Syntax (1)



'if' keyword

boolean condition to be tested

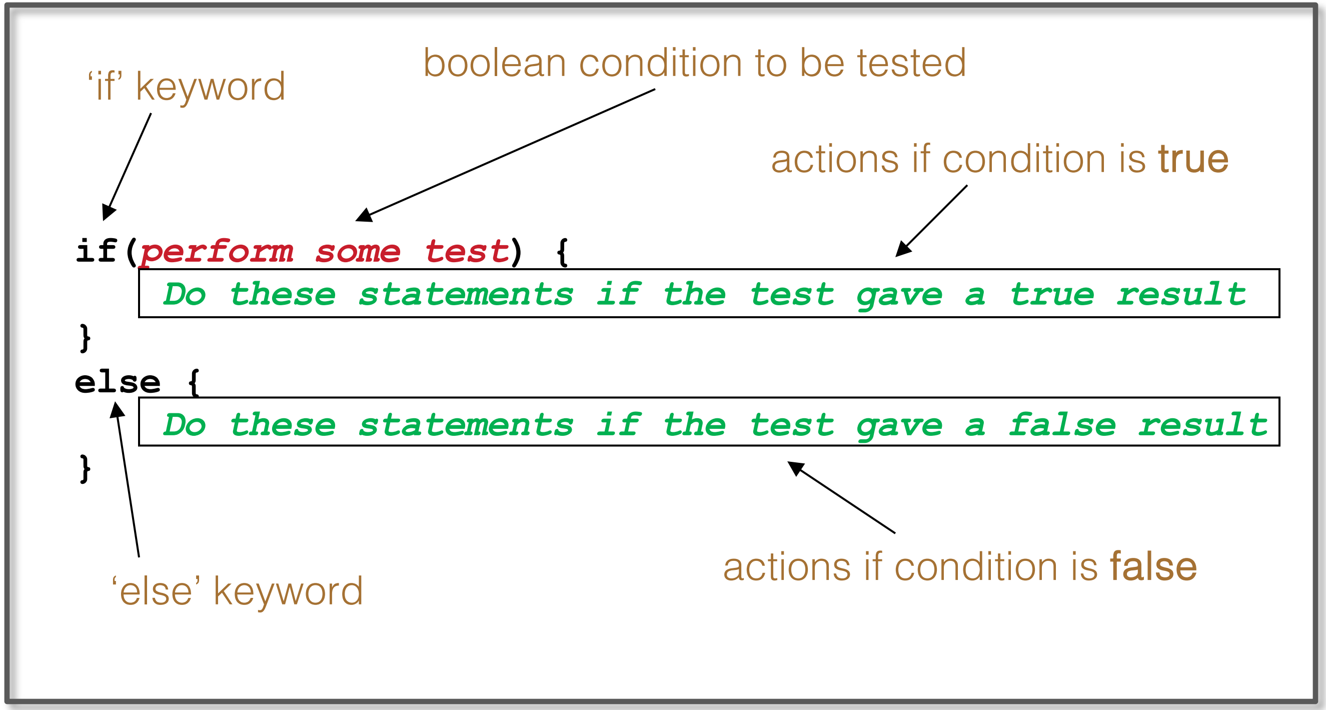
actions if condition is **true**

```
if( perform some test )
{
    Do these statements if the test gave a true result
}
```

curly
braces

parentheses

Conditional Statement Syntax (2)



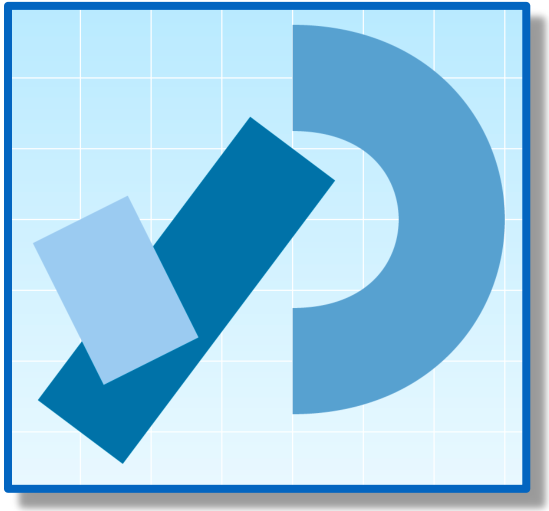
```
if(perform some test) {
    Do these statements if the test gave a true result
}
else {
    Do these statements if the test gave a false result
}
```

‘if’ keyword
boolean condition to be tested
actions if condition is **true**
‘else’ keyword
actions if condition is **false**

Conditional Statement Syntax (3)

```
if(condition1…perform some test)
{
    Do these statements if condition1 gave a true result
}
else if(condition2…perform some test)
{
    Do these statements if condition1 gave a false
    result and condition2 gave a true result
}
else
{
     Do these statements if both condition1 and
    condition2 gave a false result
}
```

Boolean Conditions & Relational Operators



Boolean conditions

- ❑ A boolean condition is an expression that evaluates to either **true** or **false** e.g.

```
mouseX < 50
```

- ❑ An if statement evaluates a boolean **condition** and its result will determine which portion of the if statement is executed.

Boolean conditions

```
// Do these statements before.

if (boolean condition)
{
      // Perform this clause if the
      // condition is true.
}

// Do these statements after.
```

Java Relational Operators

Operator	Use	Returns true if…
>	op1 > op2	op1 is greater than op2
>=	op1 >= op2	op1 is greater than or equal to op2
<	op1 < op2	op1 is less than op2
<=	op1 <= op2	op1 is less than or equal to op2
==	op1 == op2	op1 and op2 are equal
!=	op1 != op2	op1 and op2 are not equal

BEWARE = is an assignment operator.
It doesn’t test for equality. Use == to test for equality in primitive types

Source: http://www.freejavaguide.com/relational_operators.htm

Some notes on the if statement

- An if statement IS a **statement** - it is only executed once
- When your if statement only has one statement inside it, you do not need to use the curly braces

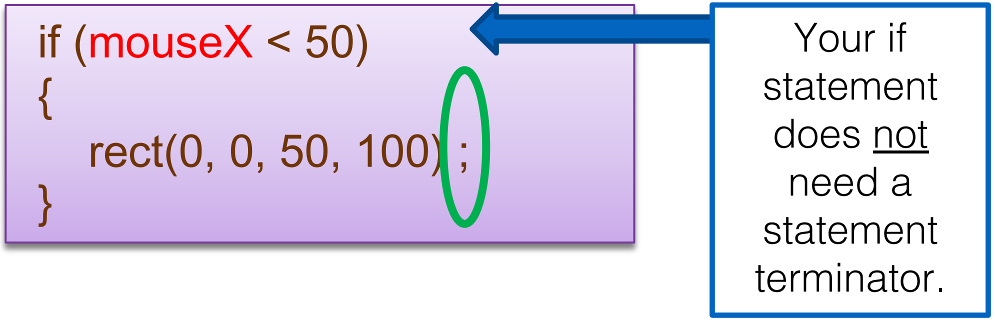
- For example, both of these are the same:

```
if (mouseX < 50)
{
    rect(0, 0, 50, 100);
}
```

```
if (mouseX < 50)
    rect(0, 0, 50, 100);
```

Some notes on the if statement

- ❑ The semi-colon (;) is a **statement terminator.**



```
if (mouseX < 50)
{
    rect(0, 0, 50, 100) ;
}
```

Your if statement does <u>not</u> need a statement terminator.

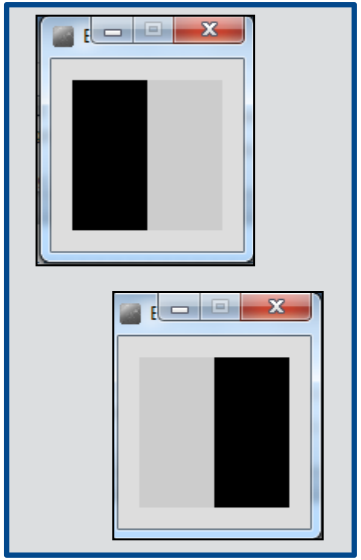
Conditional Example 3.1

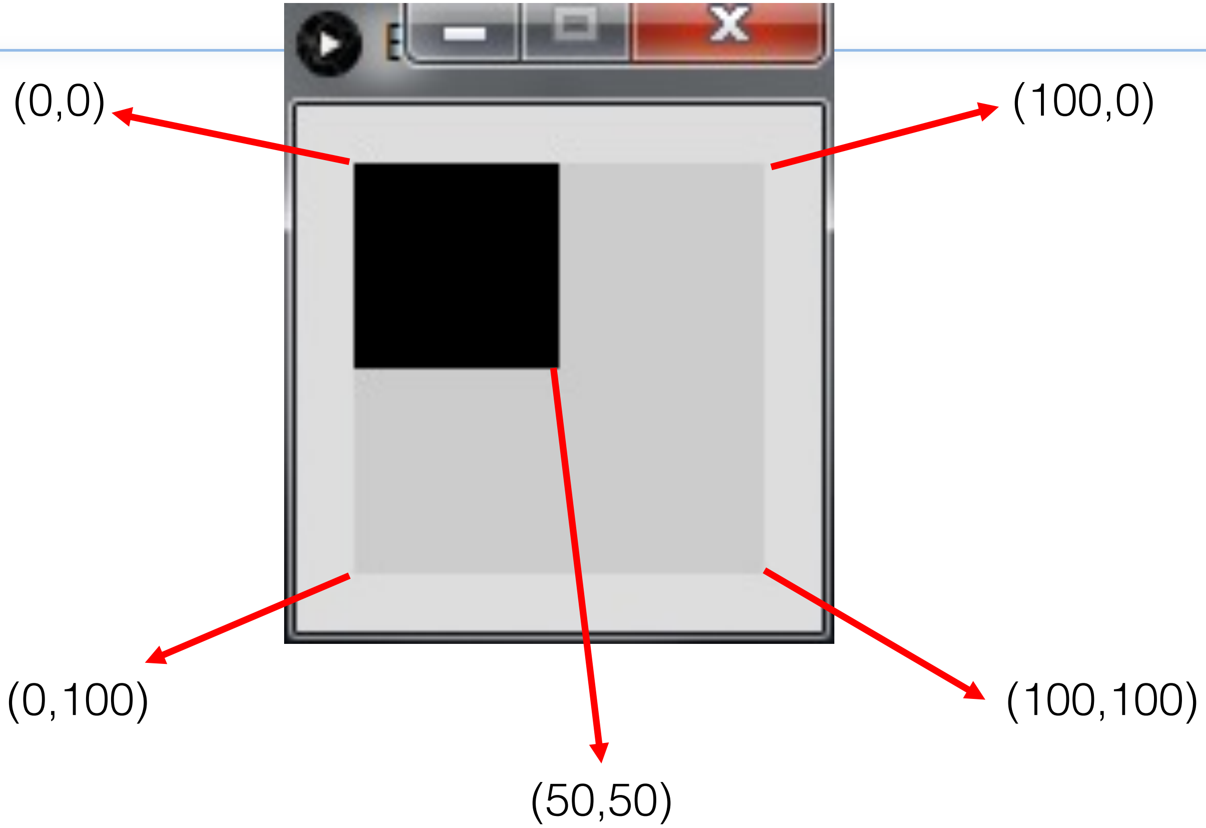
Functionality:

If the **x-coordinate of the mouse** pointer is on the:

❑ **left** half of the display window, draw a rectangle on the left hand side.

❑ **right** half of the display window, draw a rectangle on the right hand side.





(0,0)

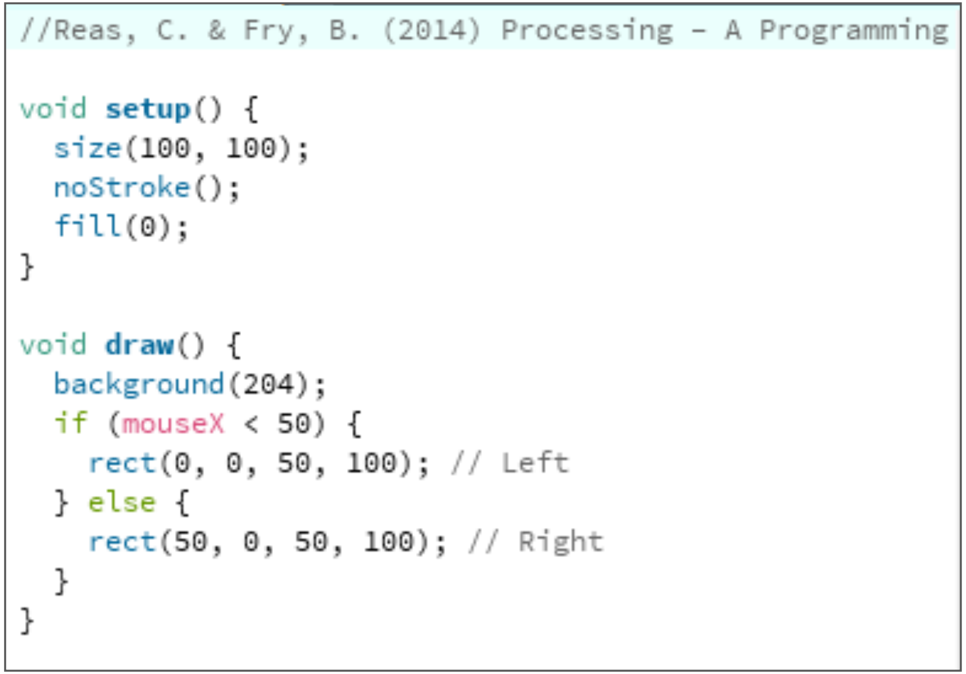
(100,0)

(0,100)

(50,50)

(100,100)

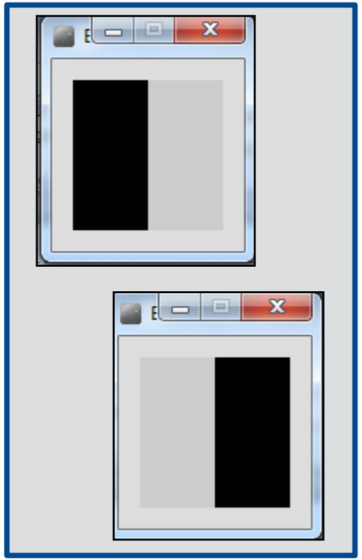
Conditional Example 3.1 - code



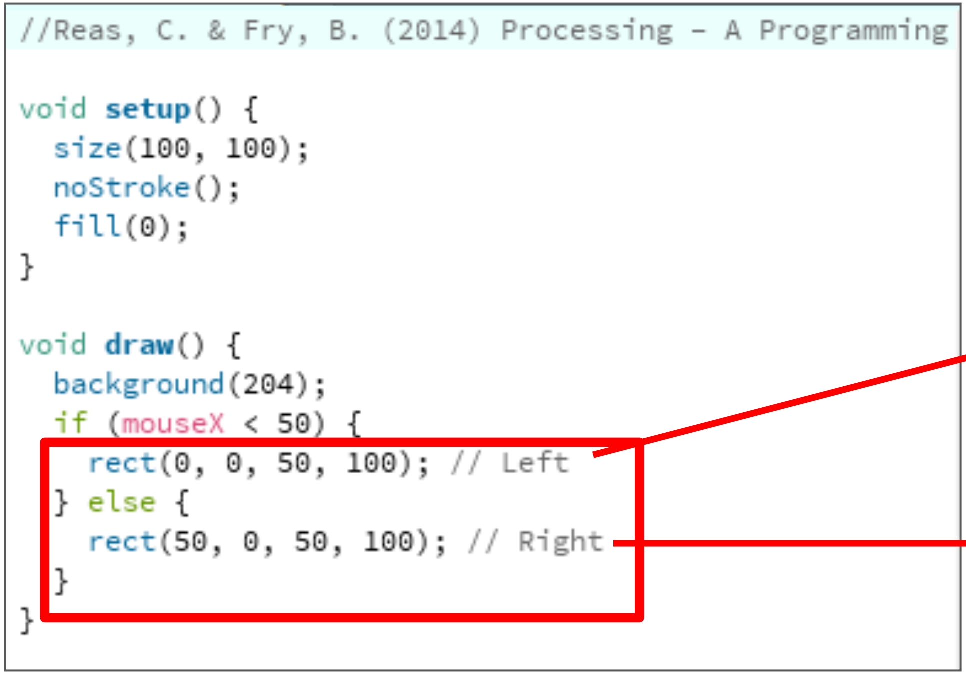
```
//Reas, C. & Fry, B. (2014) Processing – A Programming

void setup() {
  size(100, 100);
  noStroke();
  fill(0);
}

void draw() {
  background(204);
  if (mouseX < 50) {
    rect(0, 0, 50, 100); // Left
  } else {
    rect(50, 0, 50, 100); // Right
  }
}
```



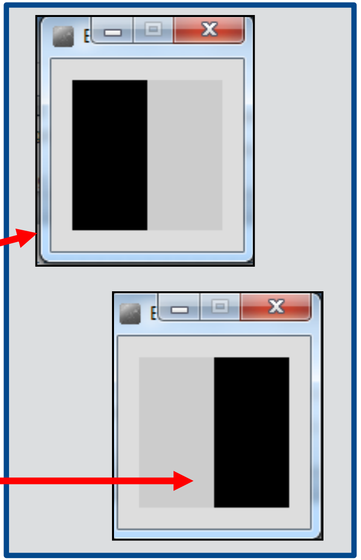
Conditional Example 3.1 - code



```
//Reas, C. & Fry, B. (2014) Processing – A Programming

void setup() {
  size(100, 100);
  noStroke();
  fill(0);
}

void draw() {
  background(204);
  if (mouseX < 50) {
    rect(0, 0, 50, 100); // Left
  } else {
    rect(50, 0, 50, 100); // Right
  }
}
```

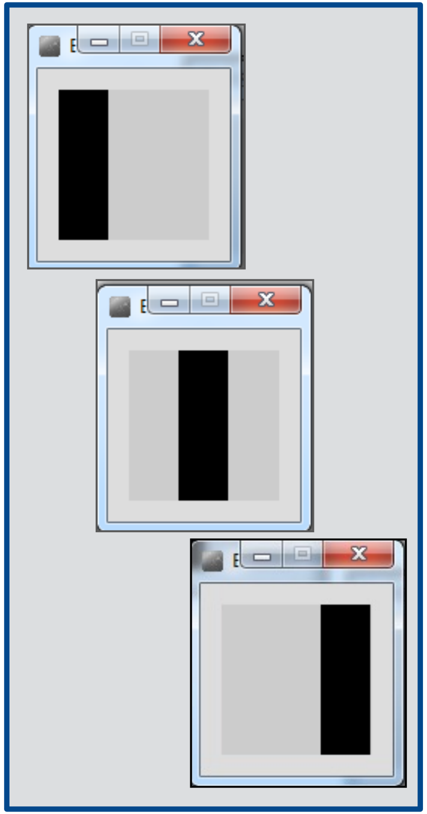


Conditional Example 3.2

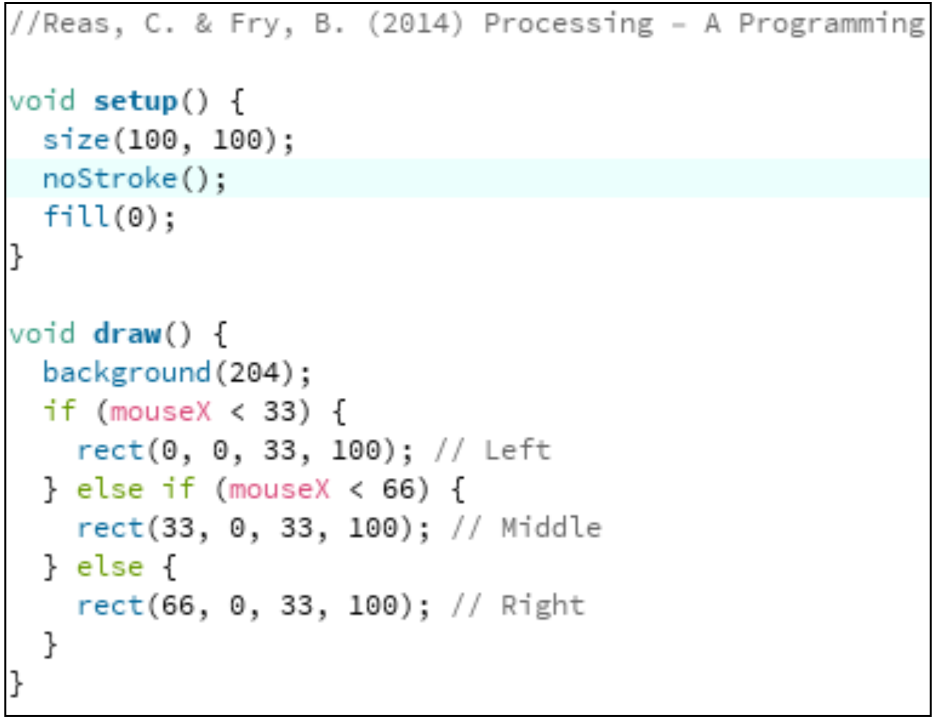
Functionality:

If the **x-coordinate of the mouse pointer** is on the:

- ❑ **left third** of the display window, draw a rectangle on the left third of the window.
- ❑ **middle third** of the display window, draw a rectangle on the middle third of the window.
- ❑ **right third** of the display window, draw a rectangle on the right third of the window.



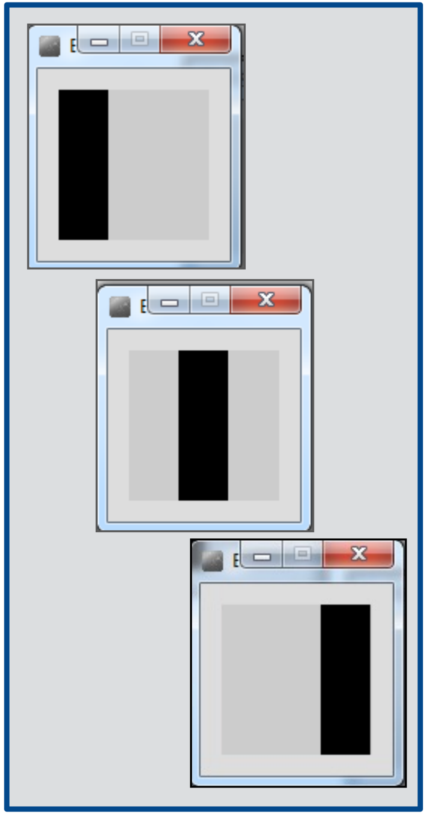
Conditional Example 3.2 - code



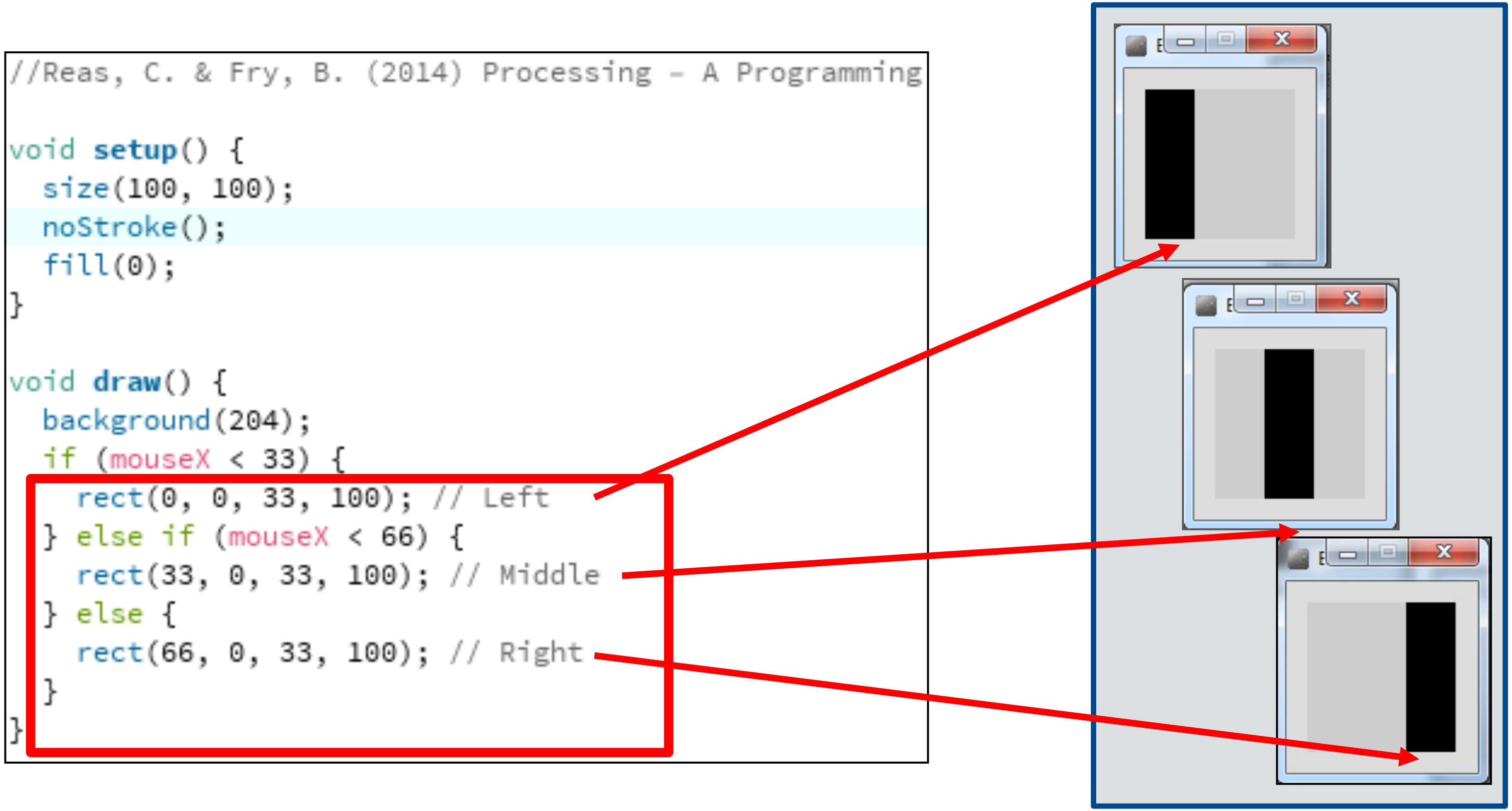
```
//Reas, C. & Fry, B. (2014) Processing – A Programming

void setup() {
  size(100, 100);
  noStroke();
  fill(0);
}

void draw() {
  background(204);
  if (mouseX < 33) {
    rect(0, 0, 33, 100); // Left
  } else if (mouseX < 66) {
    rect(33, 0, 33, 100); // Middle
  } else {
    rect(66, 0, 33, 100); // Right
  }
}
```



Conditional Example 3.2 - code



```
//Reas, C. & Fry, B. (2014) Processing – A Programming

void setup() {
  size(100, 100);
  noStroke();
  fill(0);
}

void draw() {
  background(204);
  if (mouseX < 33) {
    rect(0, 0, 33, 100); // Left
  } else if (mouseX < 66) {
    rect(33, 0, 33, 100); // Middle
  } else {
    rect(66, 0, 33, 100); // Right
  }
}
```

Logical Operators

Logical operators

- Logic operators operate on boolean values.
- They produce a new boolean value as a result.
- The ones that we will use, so far, are:

  - && (and)
  - || (or)
  - ! (not)

Logical operators - AND

a && b

- This evaluates to true if both *a* and *b* are true.
- It is false in all other cases.

a	b	a && b
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

Logical operators - OR

a || b

- This evaluates to true if either *a* or *b* or both are true, and false if they are both false.

a	b	a || b
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**

Logical operators - NOT

!a

- This evaluates to true if *a* is false, and false if *a* is true.

a	!a
0	1
1	0

Logical operators - Quiz

```
int a = 5;
int b = 10;
int c = 7;
```

What is the result of each of these boolean expressions:

Q1 (a > b) && (a < c)

Q2 (a < b) || (c < a)

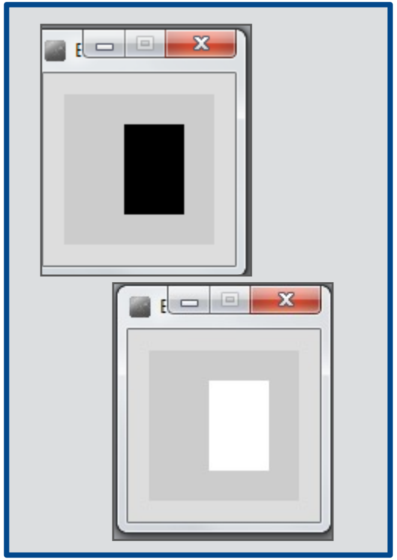
Q3 !(b < a) && (c > b)

Conditional Example 3.3

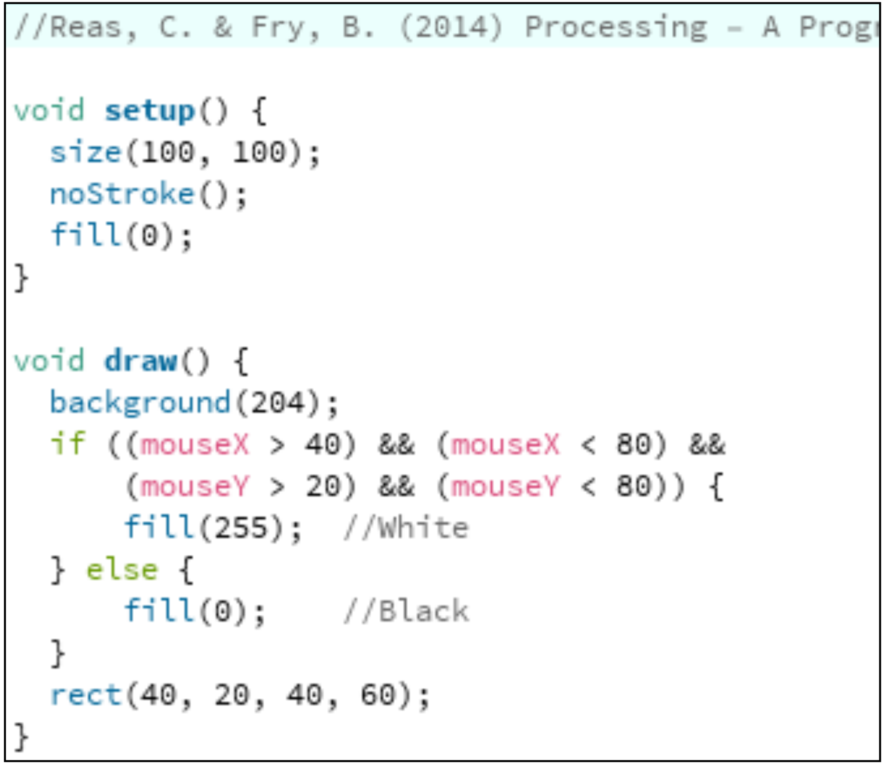
Functionality:

If the mouse pointer is:

- ❑ inside the rectangle coordinates, then fill the rectangle with white.
- ❑ otherwise, fill with black.



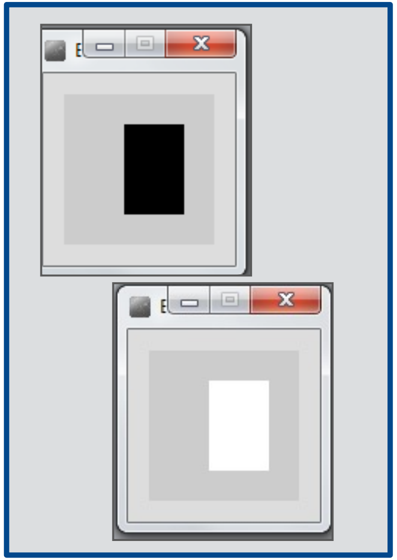
Conditional Example 3.3 - code



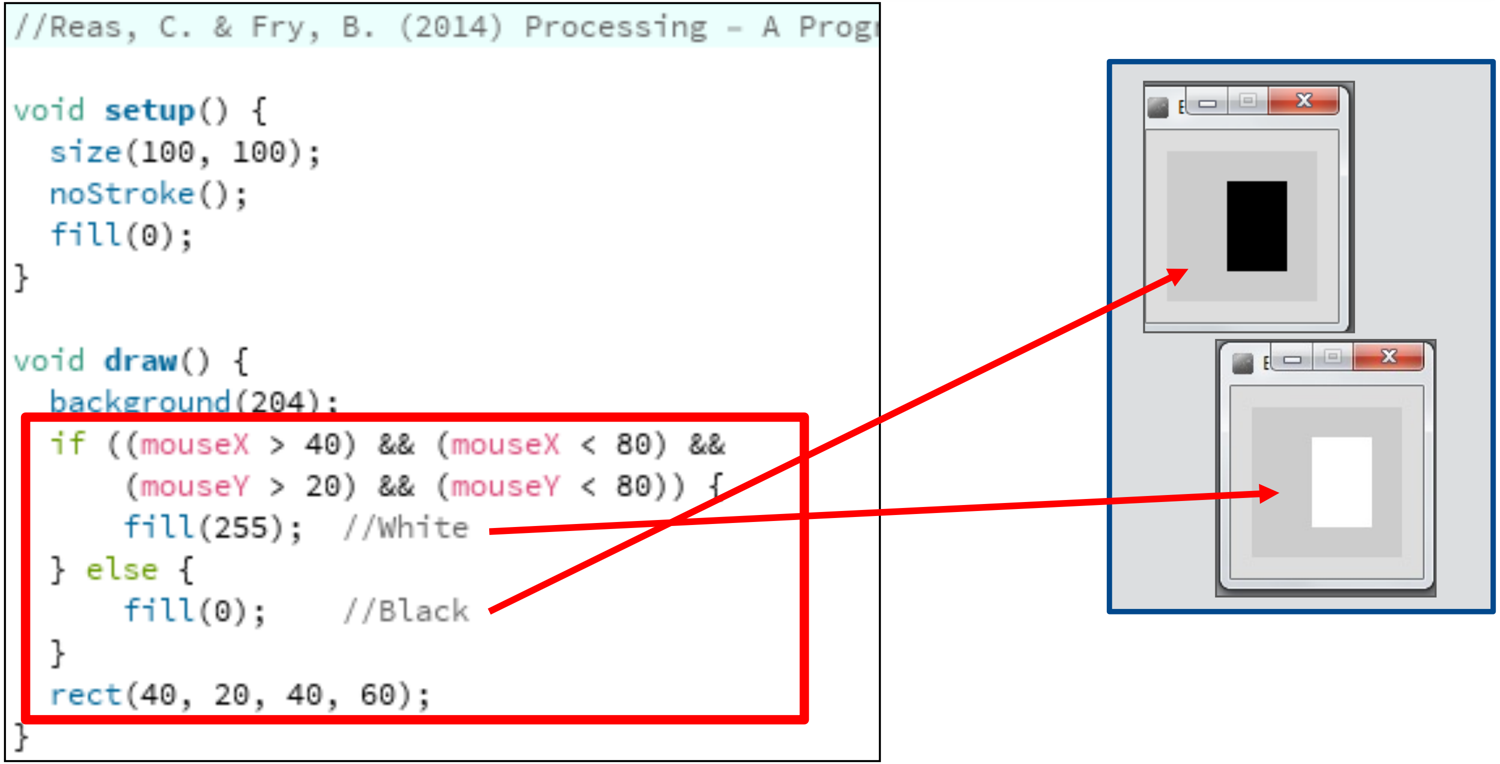
```
//Reas, C. & Fry, B. (2014) Processing - A Prog

void setup() {
  size(100, 100);
  noStroke();
  fill(0);
}

void draw() {
  background(204);
  if ((mouseX > 40) && (mouseX < 80) &&
      (mouseY > 20) && (mouseY < 80)) {
      fill(255);  //White
  } else {
      fill(0);    //Black
  }
  rect(40, 20, 40, 60);
}
```



Conditional Example 3.3 - code



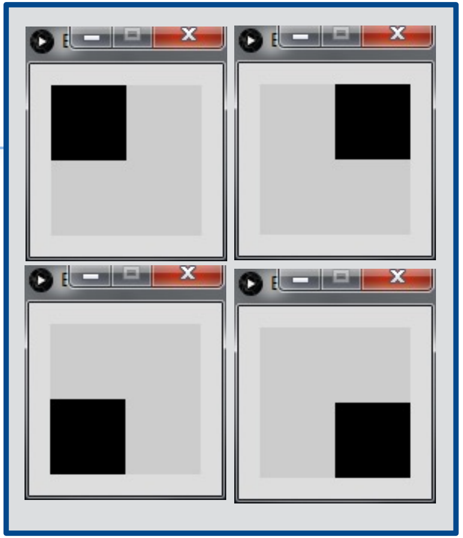
```
//Reas, C. & Fry, B. (2014) Processing - A Prog

void setup() {
  size(100, 100);
  noStroke();
  fill(0);
}

void draw() {
  background(204);
  if ((mouseX > 40) && (mouseX < 80) &&
      (mouseY > 20) && (mouseY < 80)) {
      fill(255);   //White
  } else {
      fill(0);     //Black
  }
  rect(40, 20, 40, 60);
}
```

Conditional Example 3.4

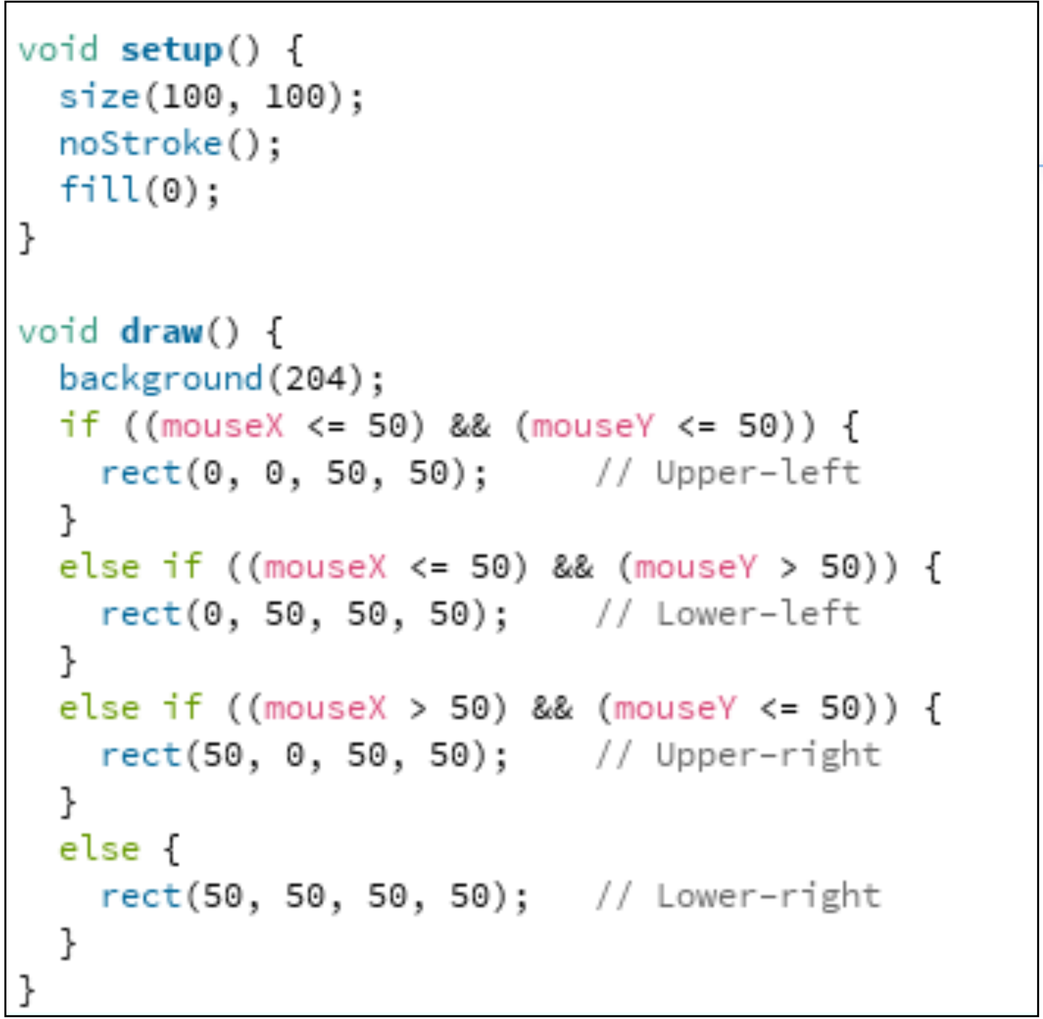
Functionality:



- ❑ If the mouse pointer is in the upper-left quadrant of the display window, draw a black rectangle covering the upper-left quadrant of the window.

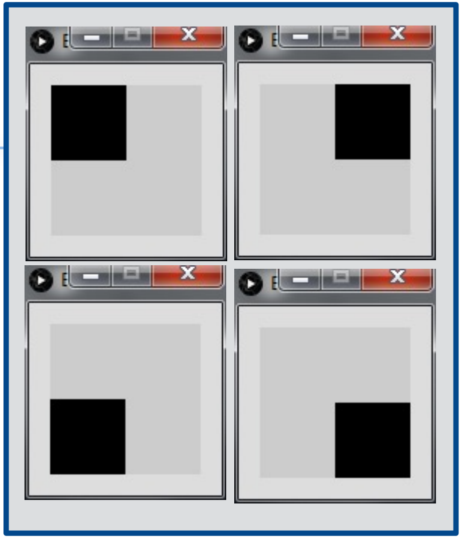
- ❑ Repeat this approach for upper-right, lower-left and lower-right quadrants.

Conditional Example 3.4 - code



```
void setup() {
  size(100, 100);
  noStroke();
  fill(0);
}

void draw() {
  background(204);
  if ((mouseX <= 50) && (mouseY <= 50)) {
    rect(0, 0, 50, 50);      // Upper-left
  }
  else if ((mouseX <= 50) && (mouseY > 50)) {
    rect(0, 50, 50, 50);     // Lower-left
  }
  else if ((mouseX > 50) && (mouseY <= 50)) {
    rect(50, 0, 50, 50);     // Upper-right
  }
  else {
    rect(50, 50, 50, 50);    // Lower-right
  }
}
```



References

- Reas, C. & Fry, B. (2014) Processing – A Programming Handbook for Visual Designers and Artists, 2nd Edition, MIT Press, London.

Questions?

Thanks.