

Conditional Events

Conditional Statements and Boolean
Expressions

Lecturer: Caio Fonseca



Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
<http://www.wit.ie/>

Topics List

- Conditional Statements
- Boolean conditions and Relational Operators
- Logical Operators

Conditional Statement Syntax (1) – if Statement

The diagram illustrates the syntax of an if statement. It shows the keyword 'if' in brown, followed by a boolean condition in red enclosed in parentheses, and a block of code in red enclosed in curly braces. Three arrows point from labels to the corresponding parts: one arrow points from the label 'if' keyword to the 'if' keyword; another arrow points from the label 'boolean condition to be tested' to the boolean condition in red; and a third arrow points from the label 'actions if condition is true' to the block of code in red.

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

Conditional Statement Syntax (2) if else Statement

```
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
}
```

Diagram annotations:

- 'if' keyword: Points to the word "if".
- boolean condition to be tested: Points to the expression "(*perform some test*)".
- actions if condition is true: Points to the block of code between the first brace and the "else" keyword.
- 'else' keyword: Points to the word "else".
- actions if condition is false: Points to the block of code between the second brace and the closing brace "}".

Conditional Statement Syntax (3) - multiple if else Statement

```
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
}
```

Conditional Statement

- Conditionals allow a program to make a decision about which lines of code to run and which not to.
- They let action take place when a specific condition is met (a test).
- The test must be an expression that resolves to true or false.
- When the expression is true the block of code inside that block will execute, if the expression is false the code is ignored.

Topics List

- Conditional Statements
- Boolean Conditions and Relational Operators
- Logical Operators

Boolean Conditions

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

$30 < 50$ (this would be true)

$60 < 50$ (this would be false)

`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
\geq	$op1 \geq op2$	op1 is greater than or equal to op2
<	$op1 < op2$	op1 is less than op2
\leq	$op1 \leq op2$	op1 is less than or equal to op2
\equiv	$op1 \equiv op2$	op1 and op2 are equal
\neq	$op1 \neq op2$	op1 and op2 are not equal

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 (30 < 50)
{
    rect(0, 0, 50, 100);
}
```

```
if (30 < 50)
    rect(0, 0, 50, 100);
```

Some notes on the if statement

- The semi-colon (;) is a statement terminator.
- One is circled in the code example below:

```
if (30 < 50)
{
    println("30 is less than 50");
}
```

- Your if statement does not need a statement terminator.

Example 1 – if Statement

The screenshot shows the Processing IDE with a sketch named "sketch_170608a". The code is as follows:

```
1 println("here is your first test");
2
3 if(30 < 50)
4 {
5     println("The test is true");
6 }
7
8 println("finished");
```

The console output at the bottom shows:

```
here is your first test
The test is true
finished
```

The test was
true so the
if block of
code **was**
executed

The test was
false so the
if block of
code **was NOT**
executed

The screenshot shows the Processing IDE with a sketch named "sketch_170608a". The code is as follows:

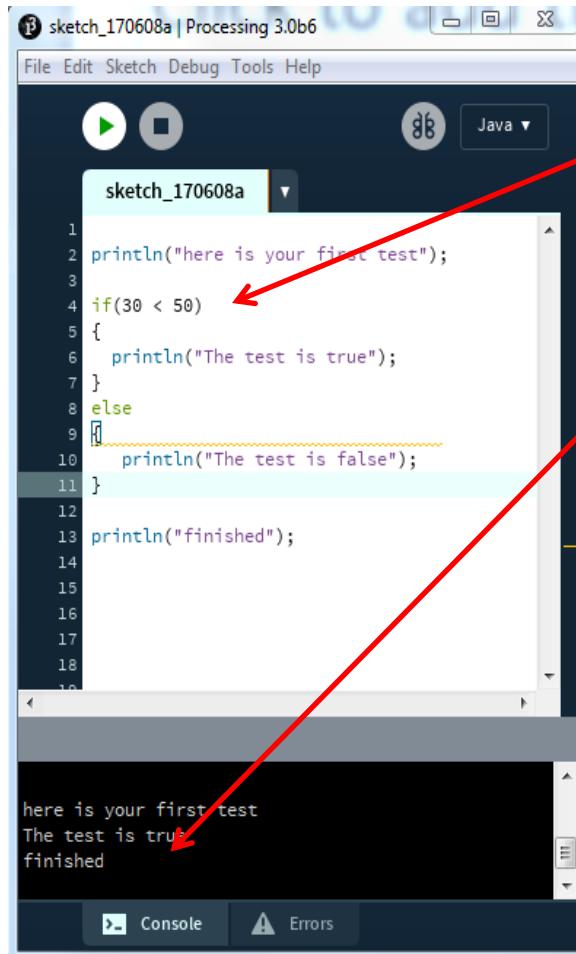
```
1 println("here is your first test");
2
3 if(60 < 50)
4 {
5     println("The test is true");
6 }
7
8 println("finished");
```

The console output at the bottom shows:

```
here is your first test
finished
```

Example 2 – if else Statement

else ignored



The screenshot shows the Processing IDE with a sketch named "sketch_170608a". The code is as follows:

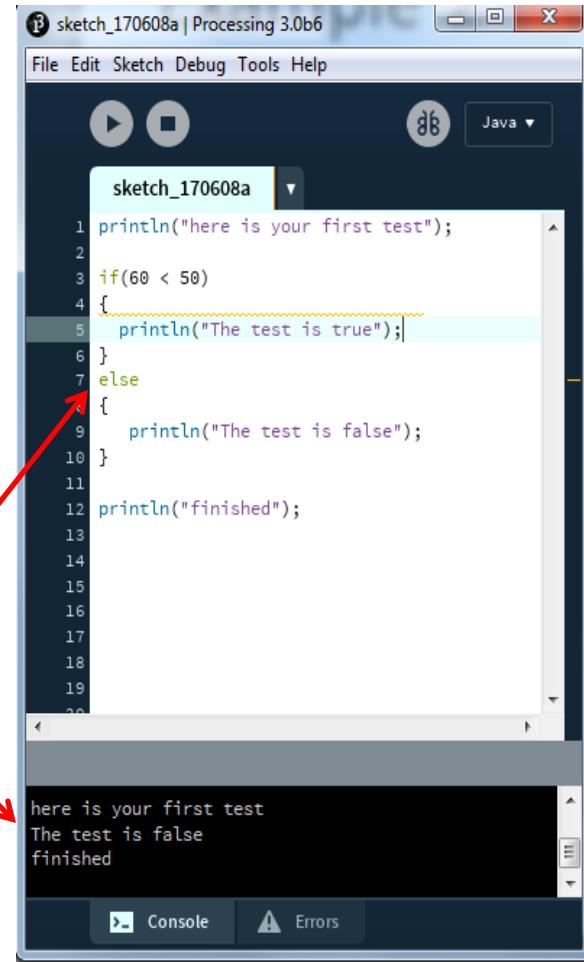
```
1 println("here is your first test");
2
3 if(30 < 50)
4 {
5     println("The test is true");
6 }
7 else
8 {
9     println("The test is false");
10}
11
12 println("finished");
13
14
15
16
17
18
19
20
```

The console output is:

```
here is your first test
The test is true
finished
```

The if part of the test was true so the if block of code was executed, the else was ignored

The if part of the test was false so it was ignored so the else block was executed.



The screenshot shows the Processing IDE with a sketch named "sketch_170608a". The code is as follows:

```
1 println("here is your first test");
2
3 if(60 < 50)
4 {
5     println("The test is true");
6 }
7 else
8 {
9     println("The test is false");
10}
11
12 println("finished");
13
14
15
16
17
18
19
20
```

The console output is:

```
here is your first test
The test is false
finished
```

if ignored

Example 3a – Multiple if else Statement

```
sketch_170608a | Processing 3.0b6
File Edit Sketch Debug Tools Help
Java ▾
sketch_170608a
1 int num = 60;
2
3 println("Here is another test");
4
5 if(num < 50)
6 {
7   println("The number is less than 50");
8 }
9 else if(num > 50)
10 {
11   println("The number is greater than 50");
12 }
13 else
14 {
15   println("They are equal");
16 }
17
18 println("You are finished");
19
20
Here is another test
The number is greater than 50
You are finished
Console Errors
```

In this example

1. the if test is true so its block of code is executed,
2. it then ignores the rest of the if else statement
3. It then jumps to the next line of code

Example 3b – Multiple if else Statement

The screenshot shows the Processing IDE interface with a Java sketch titled "sketch_170608a". The code is as follows:

```
1 int num = 50;
2
3 println("Here is another test");
4
5 if(num < 50)
6 {
7     println("The number is less than 50");
8 }
9 else if(num > 50)
10 {
11     println("The number is greater than 50");
12 }
13 else
14 {
15     println("They are equal");
16 }
17
18 println("You are finished");
19
```

A red callout box points to the first two if statements with the text: "It ignores line 5-12 as the first 2 tests are false". A red arrow points from the end of the "else if" block to the "else" block. Another red arrow points from the "else" block to the output window.

The output window displays the following text:

```
Here is another test
They are equal
You are finished
```

The "Console" and "Errors" tabs are visible at the bottom of the IDE.

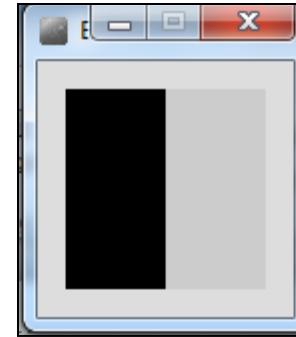
In this example

1. the if test is false so its block of code is ignored,
2. the else if test is also false so its block of code is ignored,
3. therefore the else will execute.

More Conditional Example 3.1 (if ..else)

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

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



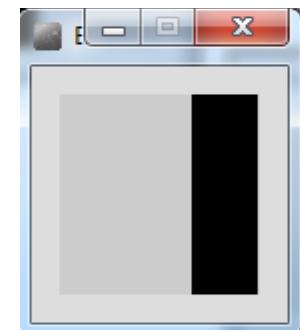
This executes
when if statement
is true



This executes
when if statement
is false

More Conditional Example 3.2 (multiple if..else)

```
void setup() {  
    size(100, 100);  
    noStroke();  
    fill(0);  
}  
  
void draw() {  
    background(204);  
    if (mouseX < 33)  
    {  
        rect(0, 0, 33, 100);  
    }  
    else if (mouseX < 66)  
    {  
        rect(33, 0, 33, 100);  
    }  
    else  
    {  
        rect(66, 0, 33, 100);  
    }  
}
```



Topics List

- Conditional Statements
- Boolean conditions and Relational Operators
- 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 are:

&& (and)

|| (or)

! (not)

Logical Operators

a && b *(and)*

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

a || b *(or)*

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

!a *(not)*

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

Logical Operators - quiz

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

What is the result of each of these boolean expressions:

$(a > b) \&\& (a < c)$

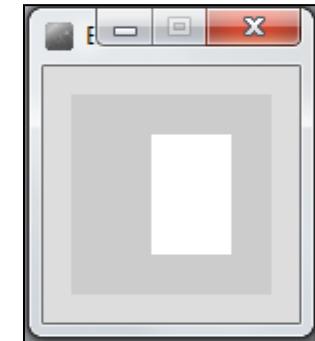
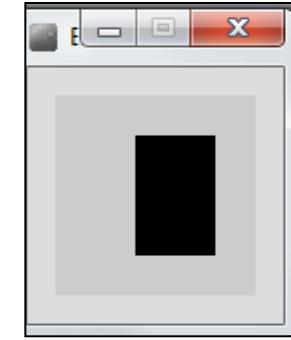
$(a < b) \mid\mid (c < a)$

$!(b < a) \&\& (c > b)$

Conditional & Logical Operators

Example 3.3

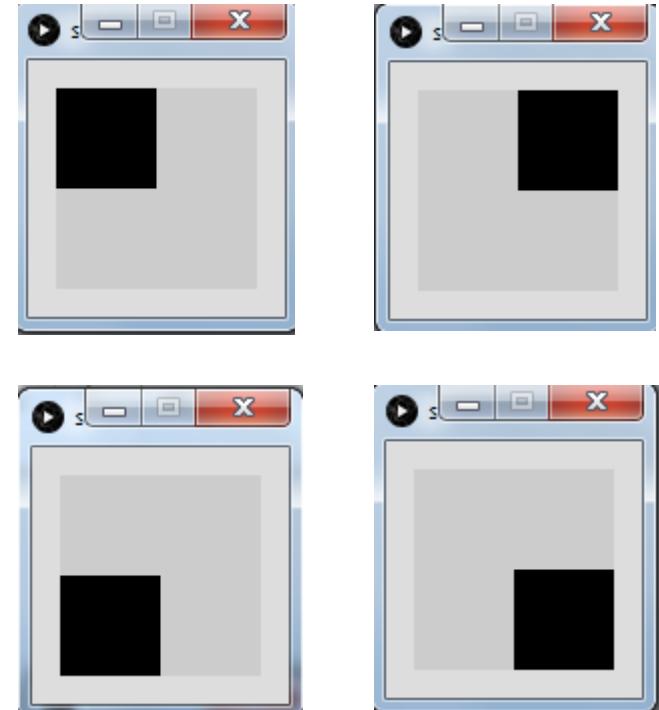
```
void setup() {  
    size(100, 100);  
    noStroke();  
    fill(0);  
}  
  
void draw() {  
    background(204);  
    if ((mouseX > 40) && (mouseX < 80) &&  
        (mouseY > 20) && (mouseY < 80))  
    {  
        fill(255);  
    }  
    else  
    {  
        fill(0);  
    }  
    rect(40, 20, 40, 60);  
}
```



Conditional & Logical Operators

Example 3.4

```
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  
    }  
}
```



Questions?





Except where otherwise noted, this content is licensed under a Creative Commons Attribution-NonCommercial 3.0 License.

For more information, please see <http://creativecommons.org/licenses/by-nc/3.0/>

Produced
by:

Dr. Siobhán Drohan
Mairead Meagher
Sinéad Walsh



Waterford Institute of Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
<http://www.wit.ie/>