

CONDITIONAL STATEMENTS

Introducing Conditional Statements:

The primary CONDITIONAL STATEMENT in JAVA is `if (boolean_expression) statement1 else statement2;`
(It is the only conditional statement covered in the AP sub-set!)

The anatomy of an if statement begins with the JAVA keyword `if`, followed by a pair of parenthesis containing a boolean expression. The boolean expression will be evaluated and if the result is `true`, the statement following the parenthesized expression (`statement1`) will be executed. If the boolean expression is evaluated as `false`, then the statement following the `else` keyword will be executed instead (`statement2`). Only one of the two statements can ever be executed, never both!

An if statement may not have an else clause.

You may have noticed that there are no curly braces in the example above!

When you write an `if` statement, it is considered to be one complete statement. If you are only doing one thing under each condition, curly braces are not necessary.

For example: (Using variable values listed below)

```
if ( x > y ) y=y*2 else x=y+1;
```

In this example `x` is greater than `y` so `y` is assigned the current value of `y` times 2 (`y` is assigned the value 10).

It is often necessary to perform multiple statements as a result of an `if` statement. In that case, you use curly braces to create a CODE BLOCK. A code block is used to group a series of JAVA statements to be executed. We have already seen this when we declare a class. All of the code is grouped into a single code block.

If we want our `if` statement to perform multiple operations, we need curly braces!

```
if ( x > y ) {  
    y=y*2;  
    x=x-y;  
} else {  
    x=y+1;  
}
```

Again, `x` is greater than `y` so `y` is assigned the current value of `y` times 2. Thus `y` is assigned the value 10. Next `x` is assigned the value of `x - y` or `17 - 10`. So `x` is assigned the value of 7.

Using the new values for `x` (7) and `y` (10), if the same if statement were executed again, what would the new values of `x` and `y` be afterwards?

Evaluating expressions as part of Conditionals:

When writing a CONDITIONAL STATEMENT you must use a Boolean expression to determine if the next line of code will be executed.

Lets look at several examples of Boolean expressions using the variables and values we have so far:

Conditional expression using relational operators:

```
if ( x > y )
```

Substituting we get:

```
if ( 17 > 5 ) substituting further if ( true )
```

So the line of code after this if statement **will** be executed.

Variable Values

Variable	Value
x	17
y	5
b	true
c	false
d	false
e	true

Conditional expression using relational operators and mathematical expressions:

```
if ( x - 2 > y * 2 )
```

Substituting we get:

```
if ( 17 - 2 > 8 * 2 )
```

 substituting further we get

```
if ( 15 > 16 )
```

 and finally we get

```
if ( false )
```

So the line of code after this if statement will **not** be executed.

Conditional expression using logical operators:

```
if ( b && e )
```

Substituting we get:

```
if ( true )
```

So the line of code after this if statement **will** be executed.

Conditional expression using relational and logical operators with mathematical expressions:

```
if ( x * 2 > x * y || c )
```

Substituting we get:

```
if ( 17 * 2 > 17 * 5 || false )
```

 substituting further we get

```
if ( 34 > 85 && false)
```

 and finally we get

```
if ( false || false )
```

So the line of code after this if statement will **not** be executed.

Variable Values	
Variable	Value
x	17
y	5
b	true
c	false
d	false
e	true

Evaluating Complete Conditional Statements:

Now let's follow a complete conditional statement example. Assume that the variables start out with the values listed above.

Line #	Code	Explanation	Variable Values
01	<pre>if (x + y < y * y) {</pre>	Using substitution we get <pre>if (17 + 5 < 5 * 5)</pre> . Substituting further we get <pre>if (22 < 25)</pre> . And finally we get <pre>if (true)</pre> so the code block starting with line 02 will be executed!	Initial Values: x = 17 y = 5
02	<pre> x = x + y;</pre>	Using substitution we get <pre>x = 17 + 5</pre> . Substituting further we get <pre>x = 22</pre> . The code block continues with line 03.	x = 22 y = 5
03	<pre> y = x / 2;</pre>	Using substitution we get <pre>y = 22 / 2</pre> . Substituting further we get <pre>y = 11</pre> .	x = 22 y = 11
04	<pre> } else {</pre>	The code block ends with line 04 and the else statement is ignored because we executed the if statement so we continue our code on line 07.	
05	<pre> y = x * 2;</pre>	Ignored!	
06	<pre> }</pre>	The end of the complete if...else statement!	
07	<pre>if (x > y * 2) {</pre>	Using substitution we get <pre>if (22 > 11 * 2)</pre> . Substituting further we get <pre>if (22 > 22)</pre> . And finally we get <pre>if (false)</pre> so the if code block starting with line 08 will be ignored.	
08	<pre> x = x * x;</pre>	Ignored because the if expression was false.	
09	<pre> } else {</pre>	Since the if expression was false, the else code block is executed starting with line 10.	
10	<pre> y = y * 2;</pre>	Using substitution we get <pre>y = 11 * 2</pre> . Substituting further we get <pre>y = 22</pre> .	x = 22 y = 22
11	<pre> }</pre>	The end of the complete if...else statement!	