Java Programming

if…then & switch

How to Write if…then statements

- Structure:

```
if something is true then
    do this
else
    do that
```

- Program statements like this are called **flow of control statements**. In Java there are two types: ***if*** and ***switch***.

if…then Statement

- An **if** statement consists of: The keyword ***if***
  - An opening parentheses *(*
  - A conditional statement
  - A closing parentheses *)*

- This may be followed by a single line of code to execute if the conditional statement is true or it may be followed by a number of statements enclosed in braces, *{}* all of which will be executed if the conditional is true.

Examples

```
if(x == 3)
    System.out.println("Bad things comes in threes!");

if(myDog == null) {
    System.out.println("Where, oh where has my little dog
  gone?");
    System.out.println("Where, oh where can he be?");
}
```

if…then…else if…else

- You can also create **if** statements in which there is code to be executed when the conditional is **false**. These statements are placed in an ***else*** block.
- For example:

```
if (x < 0)
    System.out.println("X is negative.");
else
    System.out.println("X is positive");
```

- You can also construct **if** statements in which there is more than one **if**. These are **else if** blocks.
- Here is a simple example:

```
if (x < 0)
    System.out.println("X is negative.");
else if(x == 0)
    System.out.println("X is zero.");
else
    System.out.println("X is positive");
```

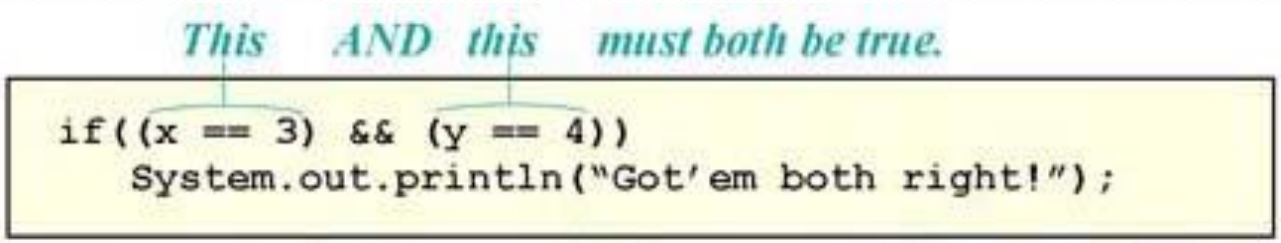
- Just as in the if block you can enclose more than one statement in the else if and the else blocks in braces {}.

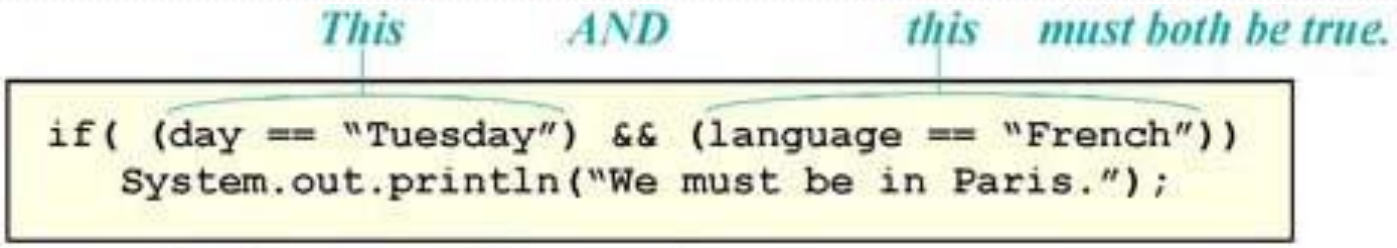
Equality and Relational Operators

- There are a number of operators you can use in **if** statements in addition to the ones described earlier:
  - **==** Equals (Yes, that is 2 equal signs)
  - **<** Less than
  - **>** Greater than
  - **<=** Less than or equal
  - **>=** Greater than or equal
  - **!=** Not equal

Conditional (Logical) && Operator

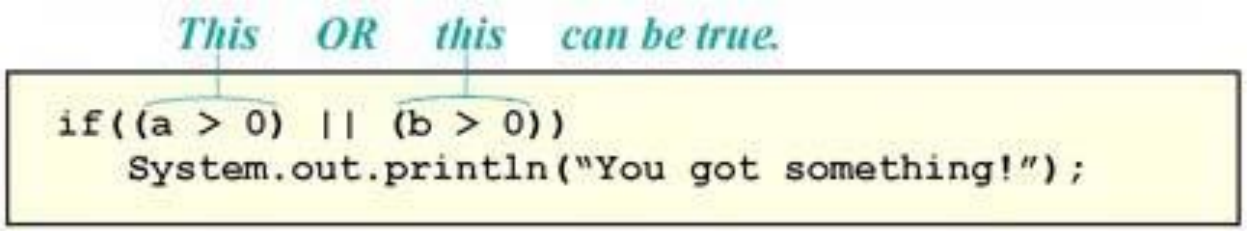
- You can construct a conditional statement with more than one item to be checked. If you want to check to see if both **x** **AND** **y** have certain values you write two conditionals and then **AND** them together using the logical **AND** operator which consists of two ampersands: ***&&***.
- Here are two examples:





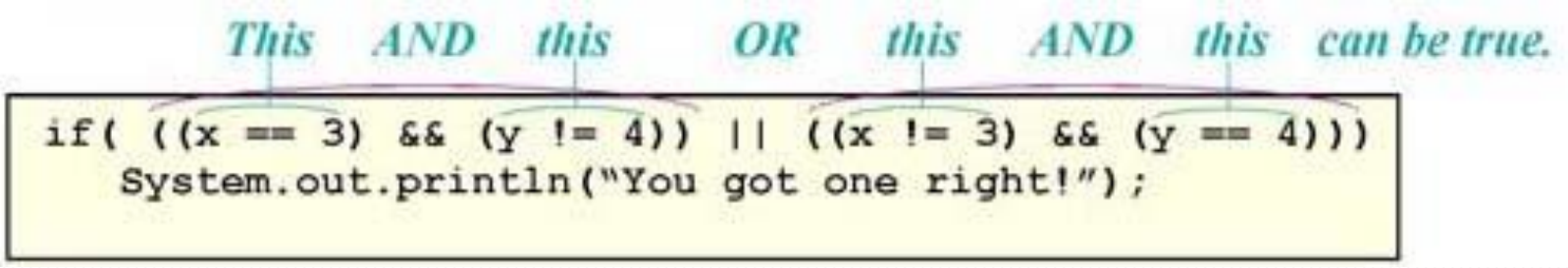
Conditional (Logical) || Operator

- If, on the other hand, you want to check to see if either of two variables has a certain value, e.g. either **a <u>OR</u> b** have certain values then you write two conditionals and **OR** them together using the logical **OR** operator which consists of two vertical bars: ||.
- Here is an example:



Complex Conditions

- In fact you can create quite complex conditionals by combining both the **AND** and the **OR** operators as shown in this example…



Conditional (Ternary) ?: Operator

- The following is a typical if..then statement:

```
int floor = 0;

if (room >= 200 && room <= 299) {
    floor = 2;
} else {
    floor = 1;
}
```

- The following evaluates the same condition using the ternary operator:

```
int floor = (room >= 200 && room < 300) ? 2 : 1;
```

switch Statement

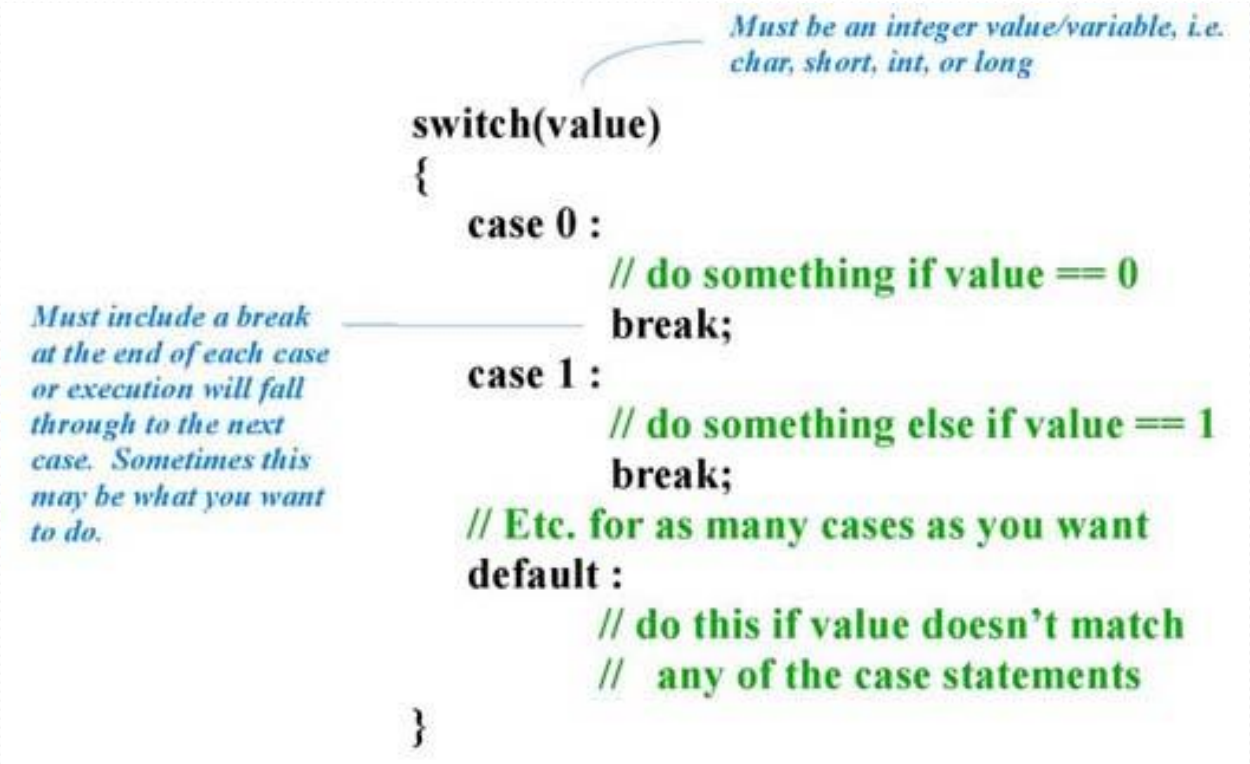
- Need to check is a variable is one of several possible values?

- Can use multiple **else…if** blocks
  - Can be tedious

- Alternate to multiple **else…if** blocks is called the **switch** statement.
  - Consists of the keyword **switch** followed by a char, byte, short, or int in parentheses.
    - Or their corresponding wrapper classes Character, Byte, Short, and Integer
    - Starting with Java SE 7, a string literal can be used as well.
  - Then in a set of braces there can be any number of **case** statements.
  - The syntax of each consists of the keyword **case** followed by a possible value for the switch variable.
  - Remember that only integer values are allowed.
  - After the value is a semicolon (i.e. :) and then all the statements to be executed if this case holds the correct value.
  - The set of statements for each case value must be ended with the keyword **break** or execution will fall through and continue with the next case statement.
  - You can also have as a "last case" a **default** statement if the none of the other **case** statements was for the current value of the variable being "switched" on.

Basic Structure

- The figure below shows the basic structure of the **switch** statement.



Example

- Below is an example of a real switch statement in code. The variable being switched on, **x**, is an **int** variable.

```
switch(x)
{
   case 0 : System.out.println("x is 0"); break;
   case 1 : System.out.println("x is 1"); break;
   case 2 : System.out.println("x is 2"); break;
   case 3 : System.out.println("x is 3"); break;
   case 4 : System.out.println("x is 4"); break;
   case 5 : System.out.println("x is 5"); break;
   case 6 : System.out.println("x is 6"); break;
   default: System.out.println("x is greater than 6");
}
```

- Sometimes you might want to take advantage of the fact that without a **break** statement at the end of a case you fall through to the next case.
- In the example below the switch variable **ans** is a **char** variable:

```
// Ask a multiple choice question and get an answer (ans)
//   as a single character (A, B, C, D) or (a, b, c, d)
switch(ans)
{
   case 'A' :
   case 'a' : // Handle answer A or a
              break;
   case 'B' :
   case 'b' : // Handle answer B or b
              break;
   case 'C' :
   case 'c' : // Handle answer C or c
              break;
   case 'D' :
   case 'd' : // Handle answer D or d
              break;
   default: System.out.println("Invalid answer");
}
```

Complex switch Statement

- The following is a slightly more complex **switch** statement in a separate program called **SwitchOff**. The **switch** statement will be executed each time through the **for** loop with different values for the loop counter **i**.

```
public class SwitchOff
{
   /** main() - the starting point.*/
   public static void main(String[] args)
   {
      // Do this in a for loop so we can check each case
      for(int i=0; i<5; i++)
      {
         // You must always switch on an integer value.
         switch(i) // Switch on the variable i
         {
            case 0 :
               System.out.println("Switch on case 0");
               break;  // Must have this or it falls through to next case
            case 1 :
               System.out.println("Switch on case 1");
               break;
            case 2 :
               System.out.println("Switch on case 2");
               break;
            case 3 :
               System.out.println("Switch on case 3");
               break;
            case 4 :
               System.out.println("Switch on case 4");
               break;
            default: // End up here if no other case matches
               System.out.println("In the default case");
         } // end of switch
      } // end for loop
   } // end main()
} // end class SwitchOff
```