

Programming technique by using java

Second class

Teaching by salma.h.abdalla

Computer Engineering & Information Technology

Lesson 16/ Thursday 11/03/2010

If statement Condition

The programming you're doing now is sequential programming, meaning the code is executed from top to bottom. It's very linear, in that each and every line of code will be read, starting with the first line of code you write and ending at the last line.

But you don't always want your programmes to work like that. Often, you want code to be executed only if certain conditions are met. For example, you might want one message to display if a user is below the age of 18 and a different message if he or she is 18 or older. You want to control the flow of the programme for yourself. You can do this with conditional logic.

Conditional logic is mainly about the IF word: IF user is less than 18 then display this message; IF user is 18 or older then display that message. Fortunately, it's very easy to use conditional logic in Java. Let's start with IF Statements.

IF Statements

Executing code when one thing happens rather than something else is so common in programming that the IF Statement has been developed. The structure of the IF Statement in Java is this:

```
if ( Statement ) {  
  
}
```

You start with the word IF (in lowercase) and a pair of round brackets. You then use a pair of curly brackets to section off a chunk of code. This chunk of code is code that you only want to execute IF your condition is met. The condition itself goes between round brackets:

```
if ( user < 18 ) {  
  
}
```

This condition says "IF user is less than 18". But instead of saying "is less than" we use the

shorthand notation of the left-pointing angle bracket ($<$). IF the user is less than 18 then we want something to happen, to display a message, for example:

```
if ( user < 18 ) {  
  
    //DISPLAY MESSAGE  
  
}
```

If the user is not less than 18 then the code between the curly brackets will be skipped, and the programme continues on its way, downwards towards the last line of code. Whatever you type between the curly brackets will only be executed IF the condition is met, and this condition goes between the round brackets.

Before we try this out, another shorthand notation is this symbol $>$. The right-pointing angle bracket means "greater than". Our IF Statement above can be amended slightly to check for users who are greater than 18:

```
if ( user > 18 ) {  
  
    //DISPLAY MESSAGE  
  
}
```

The only thing new in this code is the $>$ symbol. The condition now checks for users who are greater than 18.

But the condition doesn't check for people who are exactly 18, just those greater than 18. If you want to check for those who are 18 or over, you can say "greater than or equal to". The symbols for this are the greater than sign ($>$) followed by an equals sign ($=$):

```
if ( user >= 18 ) {  
  
    //DISPLAY MESSAGE  
  
}
```

You can also check for "less than or equal to" in a similar way:

```
if ( user <= 18 ) {  
  
    //DISPLAY MESSAGE  
  
}
```

The above code contains a less than symbol ($<$) followed by the equals sign.

Let's try all this out in a simple programme.

Start a new project by clicking **File > New Project** from the menu bar in NetBeans. You can call your package and class names anything you like. Enter the following code (our package name is **conditionallogic** and the Class is called **IFStatements**):

```
package conditionallogic;

public class IFStatements {

    public static void main(String[] args) {

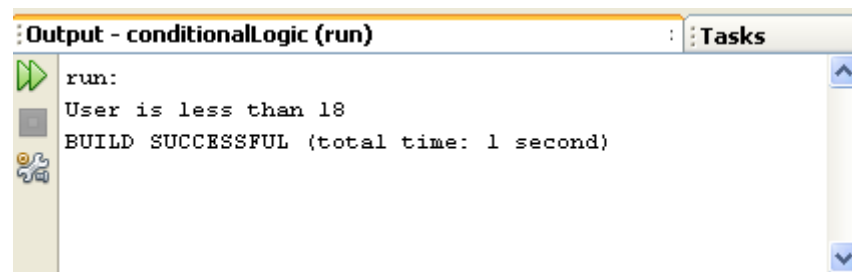
        int user = 17;

        if (user < 18) {
            System.out.println("User is less than 18");
        }

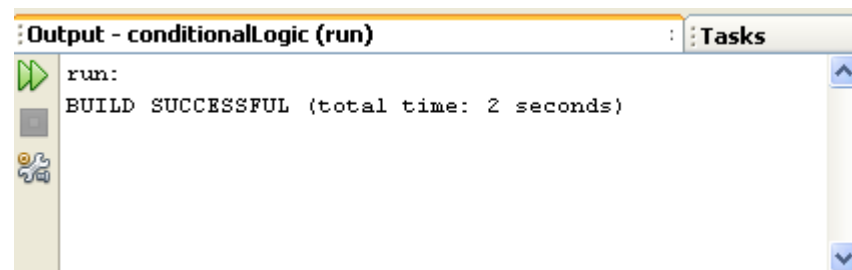
    }
}
```

We've set up an integer variable, and assigned a value of 17 to it. The IF statement checks for "less than 18". So the message between the curly brackets should be printed out.

Run your programme and check it out. (NetBeans has a habit of running the programme in bold text in the Projects window and not the code you have displayed. To run the code in your coding window, right click anywhere in the code. From the menu that appears select Run File.) You should see this in your Output window:



Now change the value for the user variable from 17 to 18. Run your programme again. You should see this:



So the programme runs OK, with no error messages. It's just that nothing gets printed out. The reason is that the message code is between the curly brackets of the IF Statement. And the IF Statement is checking for values less than 18. IF the condition is not met, Java ignores the curly

brackets altogether and moves on.

Exercise

Replace your "less than" symbol with the "less than or equal to" symbols. Change your message to suit, something like "user is less than or equal to 18". Run your programme again. Do you see the message?

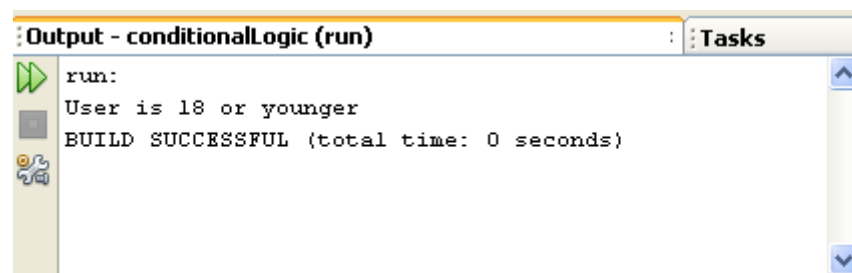
Exercise

Change the user value to 20. Run your programme again. Do you still see the message?

You can have more than one IF Statement in your code. Try the following code:

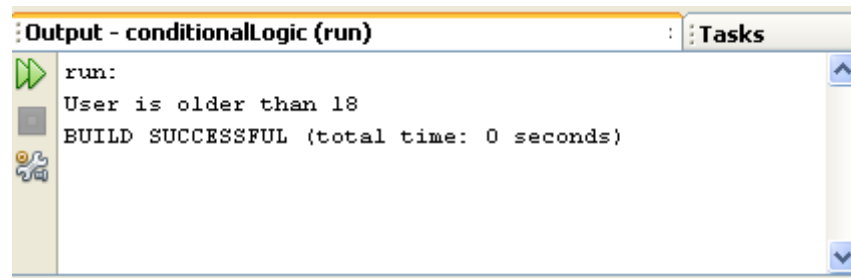
```
public static void main(String[] args) {  
  
    int user = 18;  
  
    if (user <= 18) {  
        System.out.println("User is 18 or younger");  
    }  
  
    if (user > 18) {  
        System.out.println("User is older than 18");  
    }  
  
}
```

This time, we have two IF Statements. The first tests for values less than or equal to 18. The second tests for values greater than 18. When the code is run with a value of 18 or less for the user variable, the Output is this:



```
Output - conditionalLogic (run) | Tasks  
run:  
User is 18 or younger  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Changing the value of the user variable to 20 gives this:



So only one of the IF Statements will Output a print line. And it all depends on what the value of the user variable is.

In the next part, we'll continue with Conditional Logic

Instead of using two IF Statements, you can use an IF ... ELSE Statement instead. Here's the structure of an IF ... ELSE statement:

```
    if ( condition_to_test ) {  
    }  
    else {  
    }
```

The first line starts with if, followed by the condition you want to test for. This goes between two round brackets. Again, curly brackets are used to section off the different choices. The second choice goes after the word else and between its own curly brackets. Here's our code again that checks a user's age:

```
public static void main(String[] args) {  
  
    int user = 17;  
  
    if (user <= 18) {  
        System.out.println("User is 18 or younger");  
    }  
    else {  
        System.out.println("User is older than 18");  
    }  
}
```

So there are only two choices here: either the user is 18 or younger, or the user is older than that. Adapt your code to match that in the image above and try it out. You should find that the first message prints out. Now change the value of the user variable to 20 and run the code again. The message between the ELSE curly brackets should display in the Output window.

IF ... ELSE IF

You can test for more than two choices. For example, what if we wanted to test for more age ranges, say 19 to 39, and 40 and over? For more than two choices, the IF ... ELSE IF statement can be used. The structure of an IF ... ELSE IF is this:

```
    if ( condition_one ) {  
  
    }  
    else if ( condition_two ) {  
  
    }  
    else {  
  
    }
```

The new part is this:

```
    else if ( condition_two ) {  
  
    }
```

So the first IF tests for condition number one (18 or under, for example). Next comes else if, followed by a pair of round brackets. The second condition goes between these new round brackets. Anything not caught by the first two conditions will be caught by the final else. Again, code is sectioned off using curly brackets, with each if, else if, or else having its own pair of curly brackets. Miss one out and you'll get error messages.

Before trying out some new code, you'll need to learn some more conditional operators. The ones you have used so far are these:

- > **Greater Than**
- < **Less Than**
- >= **Greater Than or Equal To**
- <= **Less Than or Equal To**

Here's four more you can use:

- && AND**
- || OR**
- == HAS A VALUE OF**
- ! NOT**

The first one is two ampersand symbols, and is used to test for more than one condition at the same time. We can use it to test for two age ranges:

```
    else if ( user > 18 && user < 40 )
```

Here, we want to check if the user is older than 18 but younger than 40. Remember, we're trying to check what is inside of the user variable. The first condition is "Greater than 18" (user > 18).

The second condition is "Less than 40" (user < 40). In between the two we have our AND operator (&&). So the whole line says "else if user is greater than 18 AND user is less than 40."

We'll get to the other three conditional operators in a moment. But here's some new code to try out:

```
public static void main(String[] args) {  
  
    int user = 21;  
  
    if (user <= 18) {  
        System.out.println("User is 18 or younger");  
    }  
    else if (user > 18 && user < 40) {  
        System.out.println("User is between 19 and 39");  
    }  
  
    else {  
        System.out.println("User is older than 40");  
    }  
}
```

Run your programme and test it out. You should be able to guess what it will print out before running it. Because we have a value of 21 for the user variable the message between the curly brackets of else if will display in the Output window.

Exercise

Change the value of the user variable from 21 to 45. The message for the else section of the code should now display.

You can add as many else if parts as you want. Suppose we wanted to check if the user was either 45 or 50. We can use two of the new conditional operators above. We can check if the user variable "has a value of 45" OR "has a value of 50":

```
else if (user == 45 || user == 50)
```

To test if the user variable has a value of something you use two equal signs, with no space between them. Java will test for that value and no other values. Because want to test for the user being 50 as well, we can have another condition in the same round brackets: user == 50. This just says "test if the user variable has a value of 50". In between the two conditions, we have the OR operator. This is two pipe characters, which is just to the left of the letter "z" on a UK keyboard. Again, there's no space between the two. The whole of the line above says "Else if the user has a value of 45 OR the user has a value of 50".

Here's our code with the new else if part:

```

public static void main(String[] args) {

    int user = 45;

    if (user <= 18) {
        System.out.println("User is 18 or younger");
    }
    else if (user > 18 && user < 40) {
        System.out.println("User is between 19 and 39");
    }
    else if (user == 45 || user == 50) {
        System.out.println("User is either 45 OR 50");
    }
    else {
        System.out.println("User is older than 40");
    }
}

```

Try it out for yourself. Change the value of the user variable to 45 and run your code. Then change it to 50 and run the code again. In both cases the new message should display.

The various conditional operators can be tricky to use. But you're just testing a variable for a particular condition. It's simply a question of picking the right conditional operator or operators for the job.

Nested IF Statements

You can nest IF Statements. (This also applies to IF ... ELSE and IF ... ELSE IF statements.) Nesting an IF Statement just means putting one IF Statement inside of another. For example, suppose you want to find out if somebody is younger than 18, but older than 16. You want to display a different message for the over 16s. You start with the first IF Statement:

```

if ( user < 19 ) {
System.out.println( "18 or younger");
}

```

To check for over 16, you can place a second IF Statement inside of the one you already have. The format is the same:

```

if ( user < 19 ) {
if ( user > 16 || user < 19 ) {
System.out.println( "You are 17 or 18");
}
}

```

So the first IF Statement catches the user variable if it's less than 19. The second IF Statement narrows the user variable down even further, for ages over 16 and under 19. To print different

messages, you can have an IF ... ELSE statement instead of the IF Statement above:

```
if ( user < 19 ) {  
  
    if ( user > 16 && user < 19 ) {  
        System.out.println( "You are 17 or 18");  
    }  
    else {  
        System.out.println( "16 or younger");  
    }  
  
}
```

Notice where all the curly brackets are in the code: get one wrong and your programme won't run.

Nested IF Statements can be tricky, but all you're trying to do is to narrow down the choices.

In the next section, you'll about the Boolean variable type.