Boolean Expressions

Now we have the ability to read in some information, calculate some formulas and display the information to the user in a nice format. However, the real power of computer programs lies in their ability to make decisions. If a computer program always executed the same exact way, it would be quite boring. Rather, we can do much more if we have "choices" to make, and based on those choices, can execute different lines of code. An if statement allows us to do this.

Before we get into the syntax of an if statement, we must discuss Boolean expressions. A Boolean expression is an expression that evaluates to true or false. For example, five is equal to two is an expression. Since we know that five equals five and not two, this expression evaluates to false. A simple Boolean expression is one that compares two values with an equality operator or a relational operator. Here is a list of both equality and relational operators:

Operator	Meaning
==	equal
!=	not equal
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to

So, for example, here are the results of several simple Boolean expressions:

Boolean Expression	Result
(35 – 7) < 28	FALSE
3 + 2*8 > 18	TRUE
88 == 33 + 55	TRUE
24 != 4*3*2*1	FALSE

In general, a simple Boolean expression has the following syntax:

<arithmetic expr.> <relational or equality operator> <arithmetic expr.>

Any of the fairly simple arithmetic expressions in the examples above can be replaced with more complex and meaningful expressions that contain variables and often operators themselves.

Boolean Operators

We can also form more complex Boolean expressions out of simple ones through Boolean operators. There are two common binary Boolean operators and one unary one. A binary operator is one which takes two operands while an unary operator takes only one operand.

Boolean operator	Meaning
&&	and
||	or
!	not

For the binary Boolean operators, the general syntax is as follows:

<Boolean expression> <Boolean operator> <Boolean expression>

The result of this Boolean operation is also Boolean (i.e. the result is true or false.)

Here are truth tables showing the result for each of these three operands:

&&	TRUE	FALSE
TRUE	TRUE	FALSE
FALSE	FALSE	FALSE

||	TRUE	FALSE
TRUE	TRUE	TRUE
FALSE	TRUE	FALSE

!	TRUE	FALSE
	FALSE	TRUE

Thus, an and operation is only true if BOTH Boolean expressions involved are true. But, an or operation is true if at least one of the Boolean expressions involved is true. Finally, the not operation negates the value of a

Boolean expression. Once again it will be instructive to look at a few examples:

Complex Boolean Expression	Value
(8 > 7) && (3 – 2 != 1)	FALSE
(4*3 <= 12) || (2 > 100)	TRUE
!(88 == (22 + 66))	FALSE

If statement

Now we are ready to see the if statement. The if statement allows you to execute a segment of code ONLY IF a particular Boolean condition is true. Here is the general syntax of an if statement:

```
if (<boolean expression>)
  stmt;
```

So, for example, here is a segment of code which prints out, “Let’s go to the beach”, if the temperature is greater than 80 degrees. (Assume that the variable temperature stores the current temperature in Fahrenheit.)

```
if (temperature > 80)
   System.out.println(“Let’s go to the beach.”);
```

Now, it is possible that we may want to execute more than one statement inside an if statement; we may want to play Frisbee if it is greater than 80 degrees outside. To include this idea, we must introduce a block of code. A block of code is designated by matching braces({}). Here is an example utilizing this idea:

```
if (temperature > 80) {
   System.out.println(“The temperature is “ + temperature + “ degrees F.”);
   System.out.println(“Let’s go to the beach.”);
   System.out.println(“Let's play Frisbee.”);
}
```

The braces indicate that everything inside of them are to be treated as the statement inside of the if. To illustrate this, a standard programming convention is to indent everything inside of the block of code uniformly, lining up the closing brace with the beginning if. We will talk more about indenting later, but the main rule you must follow is to indent everything inside of a block of code uniformly. (Notice that you also indent if there is a single statement inside of an if.)

Also, you might imagine that if the Boolean expression in question is not true, we may want to follow a different course of action. Most people enjoy the beach when it is warm outside. An if statement allows us to do

something else instead of going to the beach with an else clause. Here is the general syntax:

```
if (<boolean expression>)
   stmt1;
else
   stmt2;
```

In this construct, if the Boolean expression is true, only stmt1 gets executed, but if it is false, then stmt2 gets executed. Here is an example of such a statement:

```
if (temperature > 80) {
   System.out.println("The temperature is " + temperature + " degrees F.");
   System.out.println("Let's go to the beach.");
   System.out.println("Let's play Frisbee.");
}
else {
   System.out.println("It is chilly outside.");
   System.out.println("Therefore we must study for our Java class.");
}
```

There are a couple other variants of the if statement which we will see in the next lecture. But, for now, let's look at examples of code implementing if statements.

1) Write a program that prompts the user for a year and determines if it is a leap year or not, printing this information to the screen.

Here are the rules for a leap year:

1) It must be divisible by 4
2) It can not be divisible by 100, unless it is also divisible by 400

Here is the program:

```
// LeapYear.java
// This program prompts the user to enter a year and determines if it is a leap
```

```
// year or not.

public class LeapYear {

   public static void main(String args[]) {

      Scanner stdin = new Scanner(System.in);

      // Read in year from user.
      int year;
      System.out.println("Enter a year.");
      year = stdin.nextInt();

      // Consider case where year is divisible by 4.
      if ( year%4 == 0) {

         // Check for exception to the divisible by 4 rule.
         if (year%100 == 0 && year%400 == 0) {
            System.out.println(year + " is a leap year.");
         }
         else {
            System.out.println(year + " is not a leap year.");
         }
      }
      else {
         System.out.println(year + " is not a leap year.");
      }

   }
}
```

Now we will look at a slightly easier example. The following program lets the user convert a temperature from Fahrenheit to Celcius or vice versa:

```
// TempConv.java
// This program will let the user either convert a celsius value to a fahrenheit
// one, or the other way around.
```

```
public class TempConv {

   public static void main(String args[]) {


      Scanner stdin = new Scanner(System.in);

      char answer;
      double fahr,cel;

      // Read in user's choice of conversion.
      System.out.println("To convert Celsius to Fahrenheit enter a.");
      System.out.println("To convert Fahrenheit to Celsius enter b.");
      answer = (stdin.next()).charAt(0);

      // Execute appropriate section of code, based on user's choice.
      if (answer = 'a') {
         System.out.println("Enter a temperature in Celsius.");
         cel = stdin.nextDouble();

         // Calculate conversion and print out result.
         fahr = 32 + 1.8*cel;
         System.out.println(cel + " Celsius is " + fahr + " Fahrenheit.");
      }
      else {
         System.out.println("Enter a temperature in Fahrenheit.");
         fahr = stdin.nextDouble();

          // Calculate conversion and print out result.
         cel = (fahr – 32)*5/9;
         System.out.println(cel + " Celsius is " + fahr + " Fahrenheit.");
       }
   }
}
```