

Lecture 17

- Covers
 - Boolean expressions
- Reading: Savitch 3.4

Lecture overview

- Boolean Operators
- Precedence
- Equivalence Expressions
- Commutative and Associative Laws
- Boolean Variables: I/O and Loops

► Boolean operators

What is a logical expression?

- A logical expression evaluates to true or false
- They are most often used in if, while and do...while statements

```
if (x > 10)
{
    System.out.println("x is greater than 10");
}
```

```
while (!end)
{
    // do something
}
```

Boolean logic

- Two values true and false
- Example
 - $3 > 4$ is false
 - $3 \leq 4$ is true

Boolean operator and

- Java && operator
- Truth table for &&

e1	e2	e1&&e2
T	T	T
T	F	F
F	T	F
F	F	F

Boolean operator or

- Java `||` operator
- Truth table for `||`

e1	e2	e1 <code> </code> e2
T	T	T
T	F	T
F	T	T
F	F	F

Boolean operator not

- Java ! operator
- Truth table for !

e1	! e1
T	F
F	T

Exclusive or

- Sometimes we wish to find out if, or assert that, exactly one of two things is true, that is one is false and the other is true
- For example
 - The train is on Track 1 or Track 2
- There is no operator in Java for this exclusive form of or, but we can write an expression with that meaning using `&&` and `||`

`condition1 XOR condition2 →`

`condition1 && !condition2 || !condition1 && condition2`

Class exercise

- What is the value of the following expressions if count is 0 and limit is 10?

`(count == 0) && (limit < 20)`

`(limit > 20) || (count < -5)`

`! (count == 12)`

`(count <= 0) || (limit <= limit)`

`(count < 0) && (limit <= limit)`

► Precedence rules

Introductory example

- What is the value of this expression?
 - $9 > 6 \parallel 0 == 0 \ \&\& \ 7 == 6$
- And this one?
 - $3 + 4 * 6$
- Equivalent to
 - $3 + (4 * 6)$ // * has higher precedence

Precedence

- Java operator precedence from highest to lowest

– + - ++ -- ! (unary operators)

– * / %

– + - (binary +, - operators)

– < > <= >=

– == !=

– &&

– ||

**NB: This is not the full set of Java operators*

Precedence

- Brackets illustrating precedence applied

$9 > 6 \parallel 0 == 0 \&\& 7 == 6$

$(9 > 6) \parallel 0 == 0 \&\& 7 == 6$

$(9 > 6) \parallel (0 == 0) \&\& (7 == 6)$

$(9 > 6) \parallel ((0 == 0) \&\& (7 == 6))$

Class exercise

- What is the value of each of the following expressions where count is 0 and limit is 10?

`count == 0 && limit < 20`

`count > 0 && limit > 20 || limit > 0`

`count > 0 || limit > 20 && limit > 0`

`3 + 4 > 4 && count != 0`

Class exercise

- What is the value of each of the following expressions where count is 0 and limit is 10?

`(count == 1) && (x < y)`

`(limit < 20) || ((limit/count) > 7)`

`(count > 20) && ((limit/count) > 7)`

`(limit < 20) && ((limit/count) > 7)`

** The result of limit/count when count == 0 differs depending on whether integer or floating point division is occurring*

Evaluation

- Java uses short-circuit evaluation
- If the first part of an `||` is true, the second part of the `||` is not evaluated
- If the first part of an `&&` is false, the second part of the `&&` is not evaluated
- Example

```
int kids = 0;
```

```
if (( kids != 0) && ((pieces / kids) >= 2))
```

► Equivalent expressions

Equivalent expressions

- Some boolean expressions can be expressed in various equivalent forms
- Choose the one easiest to understand (if possible)

Relational operators (example)

- How can we decide to do something if time is not greater than limit?
 - Assume time is 20 and limit is 30
- Attempt 1

```
if (! time > limit)
{
    System.out.println("time for another game");
}
```
- Problems?

Relational operators

Attempt 2

```
if (!(time > limit))  
{  
    System.out.println("time for another game");  
}
```

- Attempt 3

```
if (time <= limit)  
{  
    System.out.println("time for another game");  
}
```

Distribution over relational operators (more examples)

$$\neg (a \geq b) \Leftrightarrow a < b$$

$$\neg (a > b) \Leftrightarrow a \leq b$$

$$\neg (a \leq b) \Leftrightarrow a > b$$

$$\neg (a < b) \Leftrightarrow a \geq b$$

$$\neg (a == b) \Leftrightarrow a != b$$

$$\neg (a != b) \Leftrightarrow a == b$$

Class exercise

- Rewrite without !

! (numberOfGames > 5)

! (balance + interest <= 2000)

! (userInput != 'q')

De Morgan's Laws

- Distribution over logical operators

$$\neg (a \ \&\& \ b) \Leftrightarrow \neg a \ || \ \neg b$$

$$\neg (a \ || \ b) \Leftrightarrow \neg a \ \&\& \ \neg b$$

Exercise

- Distribute the ! over the operators

! (apples || oranges)

! (input == 'y' && tries < 5)

! (input != 'n' && tries >= 6)

Double negation

- Two negations cancel one another

$$!!a \quad \Leftrightarrow \quad a$$

- Examples

$! (! (\text{tries} > 5))$

$!! (\text{bananas} \neq 6)$

► Commutative and Associative laws

Commutativity and associativity

- The logical operators “and” and “or” are both commutative and associative

- Commutativity

$$a \parallel b \quad \Leftrightarrow \quad b \parallel a$$

$$a \&\& b \quad \Leftrightarrow \quad b \&\& a$$

** Though be careful
with Java's short-circuit
evaluation!*

- Associativity

$$a \parallel (b \parallel c) \quad \Leftrightarrow \quad (a \parallel b) \parallel c$$

$$a \&\& (b \&\& c) \quad \Leftrightarrow \quad (a \&\& b) \&\& c$$

Class exercise

- What should the condition on the following slide be if
 - Fred always enjoys the movie if he goes on a Tuesday
 - If Fred goes with Nicola, then he enjoys the movie
 - If Fred goes with anyone but Nicola, he only enjoys the movie if he sees a science fiction or an action movie
- Assume the integer variable `day` stores the day on which he went (1 = Monday, 2 = Tuesday, etc.)
- Assume the String variable `companion` stores the name of the person with whom he went
- Assume the char variable `movieType` stores the type of movie he saw ('a' = action, 's' = science fiction, 'c' = comedy, 'r' = romance)

Class exercise

```
if ( ??? )  
{  
    System.out.println("Fred enjoyed the movies");  
}  
else  
{  
    System.out.println("Fred didn't enjoy the movies");  
}
```

Class exercise

- What should the condition on the previous slide be if the problem is changed to the following?
 - Fred always enjoys the movie if he goes on a Tuesday
 - If Fred goes with Nicola, then he only enjoys a comedy or romance
 - If Fred goes with anyone but Nicola, he only enjoys the movie if he sees a science fiction or an action movie

► Boolean variables: I/O and loops

I/O with boolean values

- Output with `System.out`

- Will output the word `true` or `false`

```
boolean b;
```

```
b = ...
```

```
System.out.println(b);
```

```
System.out.print(b);
```

- Input with the `Scanner` class

Expects `true`, `false`, (or uppercase versions)

```
b = keyboard.nextBoolean( )
```

Boolean variables to end loops

- We frequently use boolean variables to end loops
- We create and initialise a boolean variable outside the loop which is then tested in the condition of the loop
- Code must be included in the body of the loop that can set the status of the boolean variable

Example

```
boolean keepGoing = true;  
do  
{  
    // other statements  
    // If user chooses to stop, set keepGoing to false  
} while (keepGoing);
```

Next lecture

- Defining classes, attributes and methods
- Constructors
- Local variables