CONDITIONAL EXECUTION

$x > y$?

- **yes**: $\text{max} = x$
- **no**: $\text{max} = y$

<table>
<thead>
<tr>
<th>logical AND</th>
<th>logical OR</th>
<th>logical NOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;&amp;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fundamentals of Computer Science I
Outline

• Conditional Execution
  • if … then
  • if … then … else
  • Nested if … then statements
Comparisons

- Given two numbers → return a **boolean**

<table>
<thead>
<tr>
<th>operator</th>
<th>meaning</th>
<th>true example</th>
<th>false example</th>
</tr>
</thead>
<tbody>
<tr>
<td>==</td>
<td>equal</td>
<td>7 == 7</td>
<td>7 == 8</td>
</tr>
<tr>
<td>!=</td>
<td>not equal</td>
<td>7 != 8</td>
<td>7 != 7</td>
</tr>
<tr>
<td>&lt;</td>
<td>less than</td>
<td>7 &lt; 8</td>
<td>8 &lt; 7</td>
</tr>
<tr>
<td>&lt;=</td>
<td>less than or equal</td>
<td>7 &lt;= 7</td>
<td>8 &lt;= 7</td>
</tr>
<tr>
<td>&gt;</td>
<td>greater than</td>
<td>8 &gt; 7</td>
<td>7 &gt; 8</td>
</tr>
<tr>
<td>&gt;=</td>
<td>greater than or equal</td>
<td>8 &gt;= 2</td>
<td>8 &gt;= 10</td>
</tr>
</tbody>
</table>

Is the sum of a, b and c equal to 0?  
(a + b + c) == 0

Is grade in the B range?  
(grade >= 80.0) && (grade < 90.0)

Is sumItems an even number?  
(sumItems % 2) == 0
Leap Year Example
• Years divisible by 4 but not by 100 ➔ leap year
• Years divisible by 400 ➔ leap year

```java
public class LeapYear {
    public static void main(String [] args) {
        int year = Integer.parseInt(args[0]);
        boolean isLeapYear;

        // Leap year if divisible by 4 but not by 100
        isLeapYear = (year % 4 == 0) && (year % 100 != 0);

        // But also leap year if divisible by 400
        isLeapYear = isLeapYear || (year % 400 == 0);

        System.out.println(isLeapYear);
    }
}
```

% java LeapYear 2000
true
public class ArgsExample {
    public static void main(String [] args) {
        String product = args[0];
        int qty = Integer.parseInt(args[1]);
        double cost = Double.parseDouble(args[2]);
        double total = qty * cost;
        System.out.print("To buy "+ qty);
        System.out.print(" "+ product);
        System.out.println(" you will need "+ total);
    }
}
public class ComputeArea {
    /** Main method */
    public static void main(String[] args) {
        double radius;
        double area;

        // Assign a radius
        radius = 20;

        // Compute area
        area = radius * radius * 3.14159;

        // Display results
        System.out.println("The area for the circle of radius " +
                          radius + " is " + area);
    }
}
public class ComputeArea {
    /** Main method */
    public static void main(String[] args) {
        double radius;
        double area;

        // Assign a radius
        radius = 20;

        // Compute area
        area = radius * radius * 3.14159;

        // Display results
        System.out.println("The area for the circle of radius " +
             radius + " is " + area);
    }
}
public class ComputeArea {
    /** Main method */
    public static void main(String[] args) {
        double radius;
        double area;

        // Assign a radius
        radius = 20;

        // Compute area
        area = radius * radius * 3.14159;

        // Display results
        System.out.println("The area for the circle of radius " +
                          "radius + " + area);
    }
}
public class ComputeArea {
    /** Main method */
    public static void main(String[] args) {
        double radius;
        double area;

        // Assign a radius
        radius = 20;

        // Compute area
        area = radius * radius * 3.14159;

        // Display results
        System.out.println("The area for the circle of radius " +
                        radius + " is " + area);
    }
}
public class ComputeArea {
    /** Main method */
    public static void main(String[] args) {
        double radius;
        double area;

        // Assign a radius
        radius = 20;

        // Compute area
        area = radius * radius * 3.14159;

        // Display results
        System.out.println("The area for the circle of radius " + radius + " is " + area);
    }
}
Control flow

• Interesting and powerful programs need:
  • To skip over some lines
  • To repeat lines
• Conditionals → sometimes skip parts
• Loops → allow repetition of lines (we will talk about these in a future lecture)
if Statement

• Most common branching statement
  • Evaluate a boolean expression, inside the ()'s
  • If true, do some stuff
  • [optional] If false, do some other stuff

```java
if (expression)
{
    statement1;
    statement2;
    ...
}
```

Note lack of semicolon!

Curly braces used to denote a code "block":
All lines in block get executed (in sequence) or none of the them do
One-way if Statements

if (boolean-expression) {
    statement(s);
}

if (radius >= 0) {
    area = radius * radius * PI;
    System.out.println("The area + " + radius + " is " + area);
}

(A) (B)

true
false

true
false

Statement(s)

Boolean Expression

(radius >= 0)

area = radius * radius * PI;
System.out.println("The area for the circle of " + "radius " + radius + " is " + area);
Note

**if** i > 0 {
    System.out.println("i is positive");
}

(a) Wrong

**if** (i > 0) {
    System.out.println("i is positive");
}

(b) Correct

**if** (i > 0) {
    System.out.println("i is positive");
}

(a) Equivalent

**if** (i > 0) {
    System.out.println("i is positive");
}

(b) Equivalent
The Two-way `if` Statement

```java
if (boolean-expression) {
    statement(s) - for-the-true-case;
}
else {
    statement(s) - for-the-false-case;
}
```
if...else Example

```java
if (radius >= 0) {
    area = radius * radius * 3.14159;

    System.out.println("The area for the ",
                      + "circle of radius ", radius +
                      " is ", area);
} else {
    System.out.println("Negative input");
}
```
Multiple Alternative if Statements

```c
if (score >= 90.0)
  grade = 'A';
else
  if (score >= 80.0)
    grade = 'B';
  else
    if (score >= 70.0)
      grade = 'C';
    else
      if (score >= 60.0)
        grade = 'D';
      else
        grade = 'F';
```

Equivalent

```c
if (score >= 90.0)
  grade = 'A';
else if (score >= 80.0)
  grade = 'B';
else if (score >= 70.0)
  grade = 'C';
else if (score >= 60.0)
  grade = 'D';
else
  grade = 'F';
```
Trace if-else statement

Suppose score is 70.0

```java
if (score >= 90.0)
    grade = 'A';
else if (score >= 80.0)
    grade = 'B';
else if (score >= 70.0)
    grade = 'C';
else if (score >= 60.0)
    grade = 'D';
else
    grade = 'F';
```

The condition is false
Suppose score is 70.0

```java
if (score >= 90.0)
    grade = 'A';
else if (score >= 80.0)
    grade = 'B';
else if (score >= 70.0)
    grade = 'C';
else if (score >= 60.0)
    grade = 'D';
else
    grade = 'F';
```

The condition is false
Suppose score is 70.0

if (score >= 90.0)
    grade = 'A';
else if (score >= 80.0)
    grade = 'B';
else if (score >= 70.0)
    grade = 'C';
else if (score >= 60.0)
    grade = 'D';
else
    grade = 'F';

The condition is true
Suppose score is 70.0

if (score >= 90.0)
    grade = 'A';
else if (score >= 80.0)
    grade = 'B';
else if (score >= 70.0)
    grade = 'C';
else if (score >= 60.0)
    grade = 'D';
else
    grade = 'F';

grade is C
Trace if-else statement

Suppose score is 70.0

```java
if (score >= 90.0) {
    grade = 'A';
} else if (score >= 80.0) {
    grade = 'B';
} else if (score >= 70.0) {
    grade = 'C';
} else if (score >= 60.0) {
    grade = 'D';
} else {
    grade = 'F';
}
```

Exit the if statement
Note

The **else** clause matches the most recent **if** clause in the same block.

```java
int i = 1;
int j = 2;
int k = 3;

if (i > j)
    if (i > k)
        System.out.println("A");
else
    System.out.println("B");
```

Equivalent

```java
int i = 1;
int j = 2;
int k = 3;

if (i > j)
    if (i > k)
        System.out.println("A");
    else
        System.out.println("B");
```

(a)  

(b)
Nothing is printed from the preceding statement. To force the `else` clause to match the first `if` clause, you must add a pair of braces:

```java
int i = 1;
int j = 2;
int k = 3;
if (i > j) {
    if (i > k)
        System.out.println("A");
} else 
    System.out.println("B");
```

This statement prints B.
Common Errors

Adding a semicolon at the end of an if clause is a common mistake.

```java
if (radius >= 0); // Wrong
{
    area = radius*radius*PI;
    System.out.println("The area for the circle of radius "+
    radius + " is "+ area);
}
```

This mistake is hard to find, because it is not a compilation error or a runtime error, it is a logic error. This error often occurs when you use the next-line block style.
TIP

if (number % 2 == 0)
    even = true;
else
    even = false;

(b) Equivalent
    boolean even = number % 2 == 0;

if (even == true)
    System.out.println("It is even.");
(b) Equivalent
    if (even)
        System.out.println("It is even.");
### Truth Table for Operator ^

<table>
<thead>
<tr>
<th>p1</th>
<th>p2</th>
<th>p1 $^$ p2</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>false</td>
</tr>
</tbody>
</table>

**Example (assume age = 24, gender = 'F')**

- $(age > 34) ^ (gender == 'F')$ is true, because $(age > 34)$ is false but $(gender == 'F')$ is true.
- $(age > 34) || (gender == 'M')$ is false, because $(age > 34)$ and $(gender == 'M')$ are both false.
The & and | Operators

- You can use & and | instead of && and ||
- Double operators provide “lazy evaluation”
- Single operators force evaluation of all clauses in the boolean expression
The & and | Operators

If x is 1, what is x after this expression?
\((x > 1) \& (x++ < 10)\)

If x is 1, what is x after this expression?
\((1 > x) \&\& (1 > x++)\)

How about (x is 1 to start):
\((1 == x) \mid (10 > x++)?\)
\((1 == x) \mid\| (10 > x++)?\)
if statement

- `{}`'s optional if only one statement

```java
if (expression)
    statement1;

else
    statement2;
```

- Example:

```java
if (x > y)
    max = x;
else
    max = y;
```

Flowchart:

```
x > y?

  yes
    max = x;

  no
    max = y;
```
if examples

if \((x < 0)\)
\[
\begin{align*}
    &x = -x; \\
\end{align*}
\]

*Take absolute value of \(x\)*

if \((\text{Math.random()} < 0.5)\)
\[
\begin{align*}
    &\text{System.out.println("heads");} \\
\end{align*}
\]
else
\[
\begin{align*}
    &\text{System.out.println("tails");} \\
\end{align*}
\]

*Flip a fair coin and print out the results.*

if \((x > y)\)
\[
\begin{align*}
    &\begin{cases}
    &\text{int } t = x; \\
    &x = y; \\
    &y = t;
    \end{cases} \\
\end{align*}
\]

*Put \(x\) and \(y\) into sorted order*

num = 0;
if \((\text{args.length} > 0)\)
\[
\begin{align*}
    &\begin{cases}
    &\text{num = Integer.parseInt(args[0]);} \\
    \end{cases} \\
\end{align*}
\]

*If a command line option is passed in, use it as the value for num.*
Let’s Try One!!

- If the unicorn is mythical, then it is immortal, but if it is not mythical, then it is a mortal mammal. If the unicorn is either immortal or a mammal, then it is horned. The unicorn is magical if it is horned.

- Write a program named `unicorn.java` that determines whether unicorns are mythical, immortal, mortal, mammal, magical or horned, and output these to the screen. You should only output those characteristics that are true. Notice, there is no user input to this program. Your program only needs to use the above statements to determine which characteristics are true.
Summary

• Conditional Execution
  • if … then
  • if … then … else
  • Nested if … then statements