CSE11 Fall 2013
Lecture 4

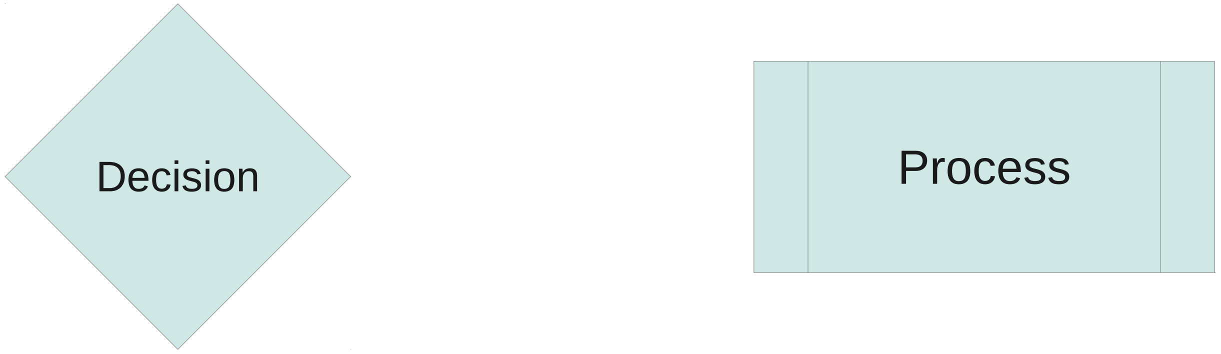
Making Choices

- Computer programs have to respond to “conditionals”
  - If (the sky is blue) then play outside
  - If (I am hungry) then eat dinner
- Must also be able to say what happens if the conditional is <u>not</u> true
  - else

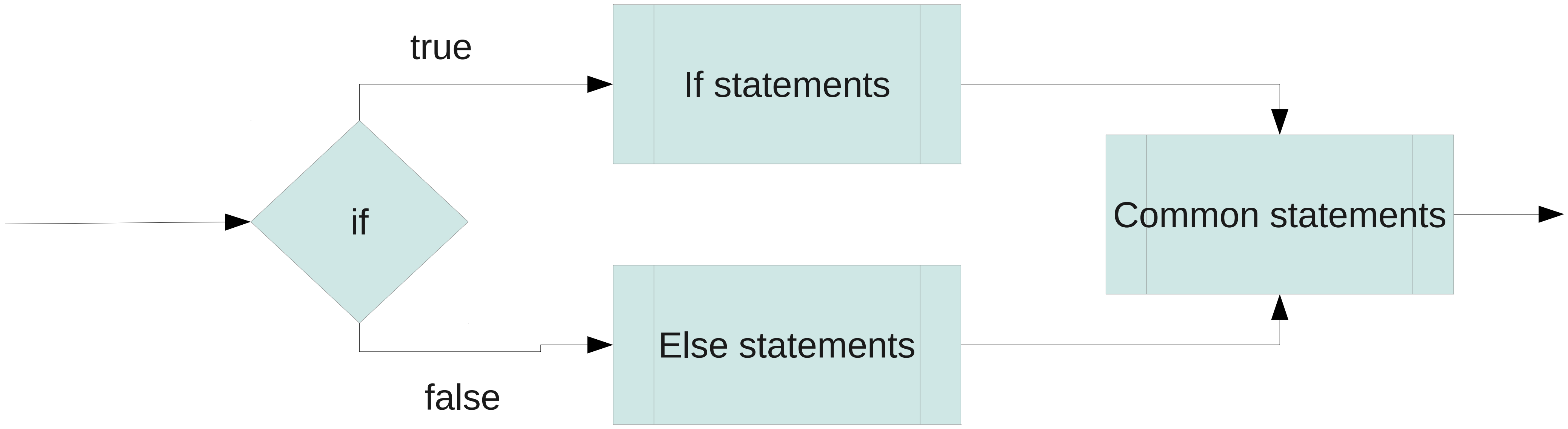
The `if` statement

```
if ( condition ) {
      if-part;  //  when condition is true
}
else  {
      else-part // when condition is not true
}
```

Flow-chart 101



This is called **Branching**
Code follows one branch or the other



Java Comparison Operators

A < B	is A *less than* B?
A > B	is A greater *than* B?
A <= B	is A *less than or Equal* to B?
A >= B	is A greater *than or Equal* to B?
A == B	is A *Equal* to B?
A != B	is A not *Equal* to B?

BEWARE!

“=” is the assignment operator
“==” is the equality comparison operator

It's very easy to confuse/misread these two

Each conditional operator evaluates to true of false

- There is no “maybe” in conditional operators
- The type is called “Boolean” (named after the 19th century Mathematician, George Boole)
- The boolean data type has only two possible values
  - True
  - False
- One can declare a variable to be of type `boolean`

Equivalent code

```java
boolean theSame;
int A;
int B;

if (A == B) {
    System.out.println("Equals")
}
```

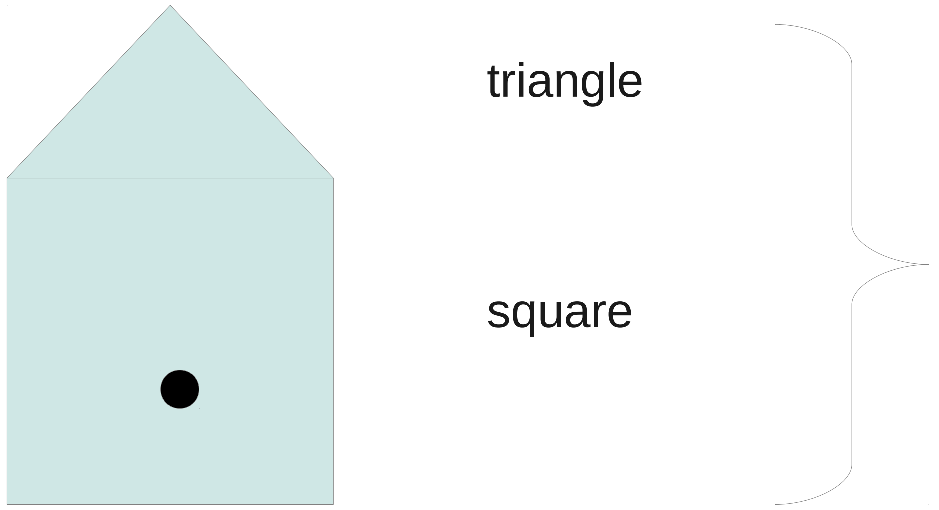
```java
boolean theSame;
int A;
int B;

theSame = A == B;
if (theSame) {
    System.out.println("Equals")
}
```

Why use boolean variables?

- Sometimes the conditions you want to test for are “complicated”.
- A suitably named boolean variable will *describe* in English the condition you want to meet
- Populating the condition (boolean) variable becomes a separate thought.

Using a Boolean Variable



Point is “in the house” if it is either in the square and/or in the triangle

```
Location point;
boolean inside;

inside = triangle.contains(point) || square.contains(point);

if (inside) {
    System.out.println(“We are warm inside!”;
}
else {
   System.out.println(“It's cold out here!”);
}
```

Java Statements/ Statement Blocks

- A java statement has a semicolon “;” at the end of it

  -
    ```
    A=25;
    Box.moveTo(30,80);
    ```
- A java statement block has an opening '{' and closing '}' with zero or more statements in it

```
{
    A=25;
    Box.moveTo(30,80);
}
{  /* empty block */ }
```

Java Expressions

- A java expression is a chunk of code the can be evaluated to be a single object
  - Most common expressions evaluate to numbers or boolean values
  - Think of an expression as a 'function' that when evaluated returns are particular value

$$\underbrace{\texttt{new filledRect(20,}\underbrace{\texttt{point.getY()}}_{\text{Expression (evaluates to a double)}}\texttt{, 5,5);}}_{\text{Java statement}}$$

The “if” construction syntax

`if ( boolean expression )` *`statement`*

`OR`

`if ( boolean expression )` *`statement-block`*

In English: This is the *syntax* of an if “statement”.

The **keyword** is **if.** Followed by parenthesis that must contain a boolean expression. Followed by a java statement OR a java statement block.

(Note, this doesn't include optional `else` or `else if` ...)

Look at WhatADrag Example from Text Book

http://eventfuljava.cs.williams.edu/sampleProgs/ch4/textbook/WhatADrag/WhatADrag.html

- Uses a Boolean variable to “communicate” between two different methods in the class
  - OnMousePress() - sets the variable boxGrabbed
  - OnMouseDrag() - drags the box only if box has already been grabbed (boxGrabbed is true)