The simple if statement

• **Program control** of the *if* statement:

```
if ( condition ) ---->
                       statement
                      +----+ FALSE
                  | condition |-----+
                  +----+
                      | TRUE
                       V
                    statement
                      V
                                  +<----+
                       Т
                       V
```

• Assembler construct that realizes the control flow of the *if* statement is:

Evaluate "condition" (CMP)			
	FALSE		
Branch on	the FALSE outcome of "condition" to here (A:)+		
	TRUE		
	V		
Translate	"statement" into assembler code		
	V		
A:	+<+		
	V		

• Example 1: find the absolute value

int x;	Assembler const	cruct for this if-statement:
if ( x < 0 ) x = -x;	MOVE.L x, D0 CMP.L #0, D0 <b>BGE L1</b>	Compares x against 0 Skip over "x=-x" when x >= 0
	MOVE.L x, D0	(you can omit this, because DO already has x)
	NEG.L DO MOVE.L DO, x	
L	l:	

The **flow chart** of the **program** is:



- Example assembler program: <u>click here</u>
- Example 2: making sure that y is the largest number

```
int x;
                     Assembler construct for this if-statement:
int y;
int help;
if ( x > y )
                     MOVE.L x, D0
                                   Compares x against y
Skip when x <= y
{ // Swap x and y CMP.L y, D0
                     BLE
  help = x;
                            L1
  x = y;
  y = help;
                     MOVE.L y, x
                                    Note: we don't need a "help"
}
                     MOVE.L D0, y
                                     variable to achieve swapping
                                      The data reg. D0 is
                                      the "help" variable...
                 L1: ...
```

The **flow chart** of the **program** is:

