**Compiling Conditional Expressions**

Expression (c ? t : e) evaluates to

* t, if c is true
* e, if c is false

[[ c ? t : e ]] =

[[ c ]]

ifeq nElse

[[ t ]]

goto nAfter

nElse: [[ e ]]

nAfter:

This is like [Compiling If Then Else](http://lara.epfl.ch/web2010/cc09:compiling_if_then_else)

* there statements did not change stack
* here t,e leave result on stack

**Example**

int f(boolean c, int x, int y) {

return (c ? x : y);

}

compiles into:

int f(boolean, int, int);

Code:

0: iload\_1

1: ifeq 8

4: iload\_2

5: goto 9

8: iload\_3

9: ireturn