

# Statistics 243: *class notes*

William J. De Meo

September 10, 1997

## Topics

### 1. Operators

- 1.1 Relational and Logical Operators
- 1.2 Increment and Decrement Operators
- 1.3 Assignment Operators

### 2. Promotion

## 1 Operators

### 1.1 Relational and Logical Operators (cont.)

Instead of

```
if(n>100){  
    if(x = getdat()){
```

one should consider using

```
if(n > 100 && x=getdat())
```

The reason: one doesn't want to have to read many if statements to understand the structure of the program.

#### Other Remarks:

The unary operator ! changes 0 to 1 and any nonzero to 0.

The assignment expression n=5 sets n equal to 5 but also evaluates to 5. So the line

```
j=(n=5)
```

assigns n equal to 5 and j equal to 5.

```
if(copy=n){ /* single equals ok */}
```

copies value of n to variable copy, then evaluates the if statements if n is nonzero. Comment is useful here since you might be tempted to correct with double equals.

## 1.2 Increment and Decrement Operators

### Some Examples

The listing

```
n=5;  
x=n++;
```

will result in `x = 5` and `n = 6`. On the other hand, the listing

```
n=5;  
x=++5;
```

results in `x = 6` and `n = 6`.

## 1.3 Assignment Operators

### Some Examples

The listing

```
b = b * scale;
```

is the same as

```
b *= scale;
```

and `i++` is the same as `i+=1` is the same as `i = i+1`.

## 1.4 Tertiary Operator

### Some Examples

The listing

```
if(x > 0) ess = x;  
else ess = 0;
```

is the same as

```
ess = x > 0 ? x : 0
```

Another commonly used example:

```
min = x > y ? y : x;
```

## 2 Promotion

The compiler will convert one value to a more meaningful value (more bits) when it encounters them in an assignment; e.g.

short  $\rightarrow$  long

float  $\rightarrow$  double

The compiler will also truncate; e.g.

```
double x;  
int ix;  
x = 7.9; ix = x;
```

yields `ix = 7`;

To convert a character representation of an integer to integer or float use: `atoi()` or `atof()`