7 If-statements

(7.1) Conditions and if-statements. An if-statement has the form (mind the **INDENTATION**)

```
  if ( <condition> )
    <statement; or {group}>

  and --- optionally ---
  else
    <statement; or {group}>
```

The condition must be in parentheses.

if (<condition>) . . .

Programming languages usually use the word ‘then.’ C doesn’t. The condition is in parentheses and ‘then’ is understood.

Statement or group of statements? It is best practice to use curly brackets *always,* as otherwise one gets into a mess. (If I forget to do so, remind me.)

```
  if ( x == 1 )
  {
    printf ("hello\n");
  }
  else
  {
    printf ("goodbye\n");
  }
```

Conditions are converted to integers. In **a.out** the condition `argc == 2` is tested and an integer produced: 1 for true and 0 for false. More generally, any integer value can be used as a condition; nonzero is treated as true and zero as false.

Complex if-statements. The basic ‘if-statement’ relations are

```
==, <, <=, >, >=, !=
```

They can be grouped into more complex statements using

```
&& for ‘and,’
|| for ‘or,’ and
! for ’not.’
```

For example, to test if a 4-digit year is a leap-year,

```
if ( yy % 400 == 0 || ( yy % 4 == 0 && yy % 100 != 0 ) )
```

Every fourth year is a leap year, except for centuries; every fourth century is a leap year.

More complex conditions can be constructed with

&& || !

for **and, or, not**. The **DOUBLE** ampersand and double bar are important; single ampersand and single bar have a different meaning.

For example, suppose yy represents a year, including the century, not just the last two digits. According to the Gregorian calendar, a leap year is

- divisible by 4, **and**
- **either is not** divisible by 100 **or is** divisible by 400.

Meaning that only one century in 4 is a leap-year; so on average the year is

$$365\frac{397}{400}$$

days long, apparently a good approximation.

This can be expressed in C:

```
... int leapyear, yy; ....

  leapyear =
    yy % 4 == 0   &&
    ( yy % 100 != 0 || yy % 400 == 0 )
  ;

  if ( leapyear ) ....
```

There are rules about the order of evaluation in the expression

```
yy % 4 == 0 &&
    ( yy % 100 != 0 || yy % 400 == 0 )
```

To be really sure, you can fully parenthesise the expression, getting

```
(yy % 4 == 0) &&
    ( (yy % 100 != 0) || (yy % 400 == 0) )
```

There are certain rules about order of evaluation, but it's hard to remember them all. Better safe than sorry.