= versus ==

The synbol = for equality was introduced by Robert Recorde in 1577, who wrote,

> And to auoide the tediouse repetition of these woordes: is equalle to: I will sette as I doe often in woorke use, a paire of parallels, or Gemove lines of one lengthe, thus: =, bicause noe .2. thynges, can be more equalle.

Recorde viewed = only as an abbreviation for the phrase "is equal to" and not as a boolean function. The concept of *function* took over 100 years to develop —Gottfried Wilhelm Leibniz introduced the term in 1694— and the notion of type *boolean* was introduced by George Boole about 150 years later, in about 1850!

Recorde's symbol for equality didn't appear in print again until 61 years later, since most authors preferred to use a word or phrase rather than a symbol for equality. But = *did* win out, partly because of its adoption by Isaac Newton and Leibniz at the close of the seventeenth century.

So, until 1969, the whole world knew that = was the equality operator. As one of our most important concepts, equality deserved its own symbol, and it had it: =.

But then, in about 1969, decisions at Bell Labs, in New Jersey, USA caused a tragic change. A few people were designing an operating system to use on their own computer in order to do research on operating systems. Bell Labs was a great place for research at that time. They needed a programming language to write the operating system, so they designed their own. The language was called C, and the operating system later became Unix!

The designers of C reasoned thus: assignment is used more often in programming than equality tests, so let's use = for assignment and == for equality. Thus was born one of the worst confusions in symbols ever to occur. It also caused great economic loss.

Mathematicians and neophtye programmers would see "x = x+1;" and ask how x = x+1 could be true —that's always false. People would write statements like

```
if (k = 5) { … }
```

in a C program. This would compile and do something, but not what they intended. It would assign 5 to k and then use 5 as the value of the expression, which in C is equivalent to true. So the condition always evaluated to true. People would spend weeks looking for the error.

This use of = for assignment and == for equality was used also in Java. To protest against this, and to make absolutely clear what is meant, Gries always writes an assignment statement with no blank before = and a blank after it, e.g.

```
x= x+1;
```

Thus, it looks non-symmetric. Since assignment is non-symmetric, it makes sense to write it this way. (x= y; is different from y= x;).

But equality is symmetric: x == y is the same as y == x . So equality is always written symmetrically, with a blank before and after ==.