

# Initialization: = or ()?

J.-C. Chappelier, V. Lepetit, J. Sam

version 1.0 of september 2013

One often encounters in the literature and in code examples the use of the equal sign for initialization, such as for example `double x =`

`3.14;` whereas in the

course we systematically opted for an initialization with parentheses: `double x(3.14);`

Is there a difference and, if so, which one? And why such a choice for the course?

If you use "basic" types, such as those presented at the beginning of this lesson (`int`, `double`, `bool`, `char`, etc.), there is no difference.

On the other hand when you use vector, string<sup>1</sup>, as for example in this course the or array "objects", or even later your own objects, then there is a subtle difference: the first (with the equal sign) is what is called an "initialization by copy" while the second (with parentheses) is "direct initialization".

Explaining the differences between these two initializations is a bit complicated, and outside the context of our course, but there are two reasons why we prefer initialization with parentheses in our course.

The first reason is simply the similarity to direct object initialization. You will thus only have one syntax to remember.

The second reason is that we want to clearly separate, especially in an introductory course, the two concepts of initialization and assignment. It seems to us clearer to use two different notations: parentheses for initialization and the equal sign for assignment.

It is for these two reasons that we advise you in this course to prefer initialization with parentheses, such as: `double x(3.14);`

Finally, note that since C++11 a third syntax, which is not presented in this course, is possible:

---

1. Objects in the sense of programming are not covered in this introductory course as such, but will be covered in our course "Introduction to Object-Oriented Programming" which will begin in February.

On the other hand, you are already going to use some without knowing it: the types `vector`, `string`, and `array` are objects.

```
double x{3.14};
```

It is still very little used, but will perhaps tend to become widespread. For my part, I do not adhere to it at all because of the confusion between on the one hand an initialization notation and on the other hand a notation for a "value of a set", such as for example

```
v = { 1, 2, 3 };
```

which is the assignment of a value to, say, a vector, as discussed later in this course.