Some Basics on C++ Variable Initialization

Alfred M. Franz

VARIABLE INITIALIZATION: EXAMPLES

Which lines of codes mean the same thing?

(T stands for any class type)

(1) `T t;`

(2) `T t();`

(3) `T t(u);`

(4) `T t = u;`

None, they are all different!

KINDS OF VARIABLE INITIALIZATION

In C++ there are three kinds of variable initialization:

1. Default initialization
2. Direct initialization
3. Copy initialization

VARIABLE INITIALIZATION: EXAMPLES

Which line is NOT an initialization?

(T stands for any class type)

(1)	`T t;`	→ default initialization
(2)	`T t();`	→ **(2) is a function declaration!**
(3)	`T t(u);`	→ direct initialization
(4)	`T t = u;`	→ copy initialization

1: DEFAULT INITIALIZATION

Example:

```
T t;
```

- code declares a variable named `t` of type `T`
- initialized using the default constructor `T:T()`

2: DIRECT INITIALIZATION

Example:

```
T t(u);
```

(assuming that u ist not the name of a type)

- code declares a variable named `t` of type `T`
- initialized directly from the value of u
- that means the `T::T(u)` is called

3: COPY INITIALIZATION

Example: `T t = u;`

- this is not an assignment
- `t` is initialized using copy constructor of `T`
- possibly after calling another function
  - if u is of type `T` it is the same as „`T t(u);`“
  - if u is of some other type it has the meaning „`T t( T(u) );`“

→ prefer using `T t(u);` instead of `T t = u;` where possible

Any Questions?

Presentation based on

[1] Herb Sutter – Exceptional C++
ISBN: 978-0-201-61562-3
Item 42: Variable Initialization—Or Is It?
Pages 175 – 176