Support de cours

Cours:

## Initiation à la programmation (en C++)

Vidéo:

## Variables (partie 3)

Concepts (extraits des sous-titres générés automatiquement) :

**Exemple d'identificateur valide. Type associé. Nombreuses situations. Moyen de la notion d'affectation. Premier caractère. Nombre de conventions. Programme des entiers positifs. Valeur de la variable. Types élémentaire principaux. Moyen de l'opérateur égal. Caractères usuelles. Droite du symbole de l'affectation. Notion de type. Nombre de mots. Type unsigned int.**

vers la recherche de séquences vidéo
(dans Initiation à la programmation (en C++).)

vers la vidéo

EPFL

# Variables
## (Partie 3)

Initiation à la programmation (C++)

Vincent Lepetit, Jean-Cédric Chappelier et Jamila Sam

...

notes

résumé

0m 0s

# Déclaration de variables

D'autres exemples de déclaration:

```
int m(1);
int p(1), q(0);
double x(0.1), y;
```

on peut déclarer plusieurs variables simultanément.
Ne pas en abuser

Il est possible en c++ de déclarer deux variables sur la même ligne.

notes

résumé

0m 1s

# Noms de variables

EPFL

**Règles pour nommer les variables:**

- Le nom doit être constitué uniquement de lettres et de chiffres (pas d'espace, ni de symboles !);
- Les accents ne sont pas autorisés;
- Le premier caractère est nécessairement une lettre;
- Le caractère souligné _ (*underscore*) est autorisé et considéré comme une lettre;
- Le nom ne doit pas être un mot-clé réservé par le langage C;
- Les majuscules et les minuscules sont autorisées mais ne sont pas équivalentes. Les noms `ligne` et `Ligne` désignent deux variables différentes.

**Exemples de noms valides:**

`n_carre` `Total` `sousTotal98`

**Exemples de noms invalides:**

`n_carré` Contient un accent;
`n carre` Contient des espaces;
`n-carre` Contient le symbole - (moins);
`1element` Commence par un chiffre.

Dans ce cas, on n'indique qu'une seule fois le type associé, et on sépare les différentes déclarations par une virgule. Ceci peut donner lieu à des ambiguïtés, il est conseillé de ne pas en abuser. Il existe un certain nombre de conventions à respecter lorsqu'on déclare une variable, lorsqu'on donne un identificateur à une variable. Il faut que l'identificateur soit constitué de lettres et de chiffres, sachant que le caractère souligné est considéré comme une lettre. Vous avez ici un exemple d'identificateur valide. Les accents ne sont pas autorisés. Vous avez un exemple ici d'identificateur invalide, contenant un accent. De même, le premier caractère doit être une lettre, et vous avez ici un exemple d'identificateur invalide, commençant par un chiffre.

notes

résumé

0m 6s

# Types de variables

Les principaux types élémentaires sont:

- `int`, pour les valeurs entières (pour *integer*, entiers en anglais);
- `double`, pour les nombres à virgule, par exemple 0.5

et aussi:

- `unsigned int`: pour les entiers positifs;
- `char`: pour les caractères (A..Z etc.);
- ...

Bien sûr, l'identificateur ne doit pas être un mot réservé du langage, il existe un certain nombre de mots réservés qu'il convient de ne pas utiliser comme nom de variable. Les majuscules et minuscules sont autorisées mais ne sont pas équivalentes, ce qui signifie par exemple, qu'un identificateur comme ligne ou Ligne ne sont pas équivalents. Ils ne vont pas désigner la même variable. Nous avons vu la notion de type, qui est essentielle pour déclarer des variables. Les deux types élémentaire principaux qui nous permettent dans un programme de manipuler des donnés numériques, de déclarer des variables de type numérique, sont int et double, que nous avons déjà croisées dans divers exemples. Il existe bien sûr beaucoup d'autres types prédéfinis en c++. Par exemple, si je veux manipuler dans un programme des entiers positifs, par exemple si je veux représenter le nombre d'étudiants suivant ce cours, je peux avoir recours au type unsigned int, qui précisément veut dire entier positif, et qui va me permettre de caractériser ma donnée de façon plus précise, que uniquement par le biais d'un int. De même, si j'ai besoin dans un programme de manipuler des caractères,

notes

résumé

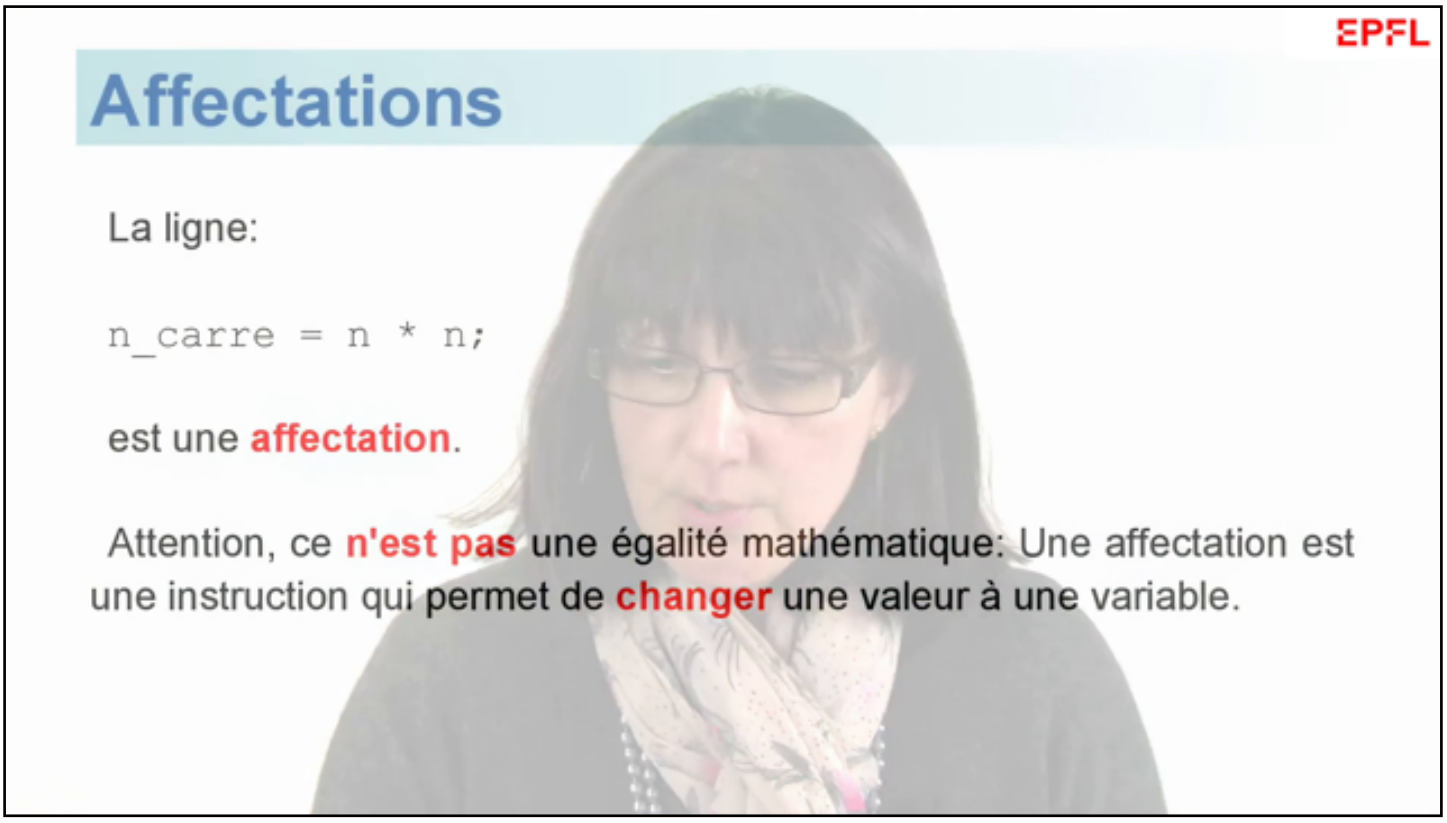
1m 1s


les caractères usuelles entre a et z, je peux avoir recours au type char. Nous aurons l'occasion lors de ce cours de revenir sur des types prédéfinis utiles dans le langage c++.

notes

résumé

2m 13s




Nous savons à ce stade déclarer et initialiser une variable, pour y stocker une donné. Dans de nombreuses situations, il faudra changer la valeur de la variable, en cours d'exécution du programme.

notes

résumé

2m 25s

# Affectations

La ligne:

```
n_carre = n * n;
```

est une **affectation**.

Attention, ce **n'est pas** une égalité mathématique: Une affectation est une instruction qui permet de **changer** une valeur à une variable.

Ceci se fait au moyen de la notion d'affectation. L'affectation se pratique au moyen de l'opérateur égal, dit opérateur d'affectation. Une ligne de cette nature se lit, j'affecte à la variable n carré, la nouvelle valeur, n fois n.

notes

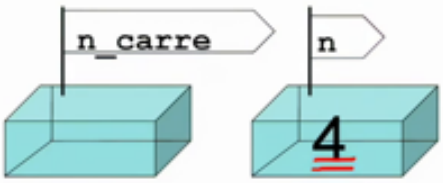
résumé

2m 36s

# Affectations

L'exécution d'une affectation se décompose en deux temps :

```
n_carre = n * n;
```




Ce qui signifie que je suis en train de changer la valeur de la variable, pour y stocker la nouvelle valeur n fois n. Attention à ne pas confondre avec une égalité mathématique. L'exécution d'une instruction d'affectation se déroule en réalité en deux temps. Dans un premier temps, ce qui est à droite du symbole de l'affectation, à droite du symbole égal, va être évalué. Ici, dans notre exemple, nous avons une variable n qui stocke la valeur quatre. Sachant qu'en c++, l'étoile représente la multiplication,

notes

résumé

2m 49s

# Affectations

De façon plus générale, une affectation suit le schéma:

```
nom_de_variable = expression;
```

Une expression calcule une valeur, qui doit être de même type que la variable.

Exemples d'expression:

- `4`
- `n * n`
- `n * (n + 1) + 3 * n - 2`

Nous reviendrons sur les expressions un peu plus loin.

le résultat de l'évaluation de n fois n, retourne tout simplement 16. Dans un second temps, ce que nous avons évalué à l'étape une, va être tout simplement stocké dans la variable qui est à gauche du symbole d'affectation. Ici, désormais la variable n carré stocke la valeur 16. Il faut bien réaliser ici, que si au préalable, n carré contenait autre chose que 16, ce résultat est effacé et remplacé par la nouvelle valeur, 16. De façon plus générale, l'affectation suit le schéma suivant. Vous avez un nom de variable suivi du symbole d'affectation égal,

notes

résumé

3m 25s

et suivi d'une expression qui va déterminer la nouvelle valeur que va prendre la variable, suite à l'affectation. Nous allons bien sûr voir les expressions plus en détails plus loin. Vous en avez un certain nombre ici. Une expression peut se réduire à une simple valeur élémentaire, une expression simple, comme ici le calcul du carré de n. Et on utilisera les opérateurs arithmétiques usuels, que l'on verra plus tard, la multiplication, le moins... Bien sûr, ne pas oublier le fameux petit point virgule à la fin, qui est essentiel pour que la compilation se passe de façon satisfaisante. se passe de façon satisfaisante.

notes

résumé

4m 1s