

Support de cours

Cours:

Initiation à la programmation (en C++)

Vidéo:

Variables - lecture, écriture (partie 4)

Concepts (extraits des sous-titres générés automatiquement) :

Variable n. Programme d'exemples. Lecture de valeurs de variables. Instruction cin. Variable b. Variable. Flot d'entrée standard. Lecture d'une variable. Valeur. Variable b de type entier. Instruction cout. Signe similaire. Suite du programme. Variable intermédiaire. Nom d'une variable.



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



[vers la vidéo](#)

Center for Digital Education. Plus de matériel de soutien pédagogique ici :

<https://www.epfl.ch/education/educational-initiatives/cede/educational-technologies-gallery/boocs-en/>
page 1/13

Variables : lecture/écriture

(Partie 4)

Initiation à la programmation (C++)

Vincent Lepetit, Jean-Cédric Chappelier et Jamila Sam

...

notes

résumé

0m 0s



Réponse

```
int a(1);
int b(2);

a = b;
b = a;

cout << a << ", " << b << endl;
```

A: 1, 1

B: 1, 2

C: 2, 2

D: 2, 1



La réponse : déroulons-la pas à pas de nouveau.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

.....

.....

.....

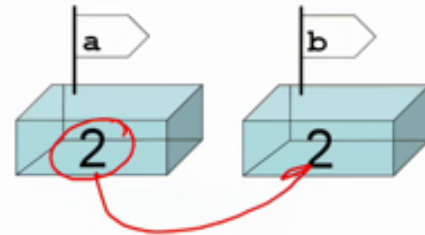
.....

.....

0m 1s



```
int a(1);  
int b(2);  
  
a = b;  
b = a;  
  
cout << a << ", " << b << endl;
```



On commence par déclarer une variable a que l'on initialise à la valeur 1, puis on déclare une variable b de type entier qu'on initialise à la valeur 2, ensuite on recopie la valeur de b dans a, donc ici, on va avoir a qui vaut 2, puis on recopie la valeur de a qui a été modifiée,

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

.....

.....

.....

.....

.....

0m 5s



Supposons qu'on ait déclaré et initialisé deux variables a et b .

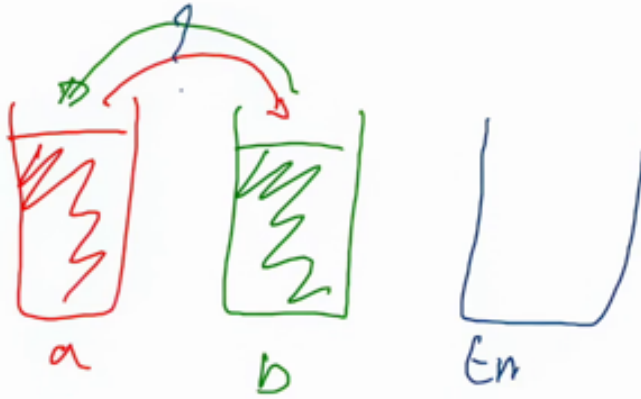
Comment échanger leurs valeurs ?

Les instructions:

$a = b;$

$b = a;$

ne marchent pas, comme le montre la question précédente.



on recopie la valeur de a dans b , reste 2, et donc au final, on va afficher 2, 2. La question qu'on pourrait se poser, c'est comment échanger la valeur de deux variables ? Supposons qu'on ait une variable a qui au départ contient 1, une variable b qui au départ contient 2, comment fait-on pour que l'on ait, après échange, un 2 dans la variable a et un 1 dans la variable b ? Supposons que vous ayez à le faire avec des verres à eau. Supposons qu'on ait ici un verre a rempli de grenadine et un verre b rempli de menthe. Comment feriez-vous, concrètement, pour transférer la menthe dans le verre a et transférer la grenadine dans le verre b ? La solution consiste bien sûr à utiliser un troisième verre

notes

résumé

0m 31s



```

#include <iostream>
using namespace std;

int main()
{
    int n(4);
    int n_carre;

    n_carre = n * n;

    cout << "La variable n contient " << n << "." << endl;
    cout << "Le carre de " << n << " est " << n_carre << "." << endl;
    cout << "Le double de n est " << 2 * n << "." << endl;

    return 0;
}

```

que je vais appeler tmp, ce n'est pas plus compliqué pour des variables, il suffit simplement de penser à utiliser, à introduire, une variable intermédiaire. Ce que l'on va faire, on va commencer par créer cette variable intermédiaire temp dans laquelle on va recopier la valeur de a. Donc on va initialiser temp avec la valeur de a, donc ici, temp vaudra 1. Puis on pourra recopier b dans a. Enfin, on pourra recopier la valeur de temp dans b. Passons maintenant à la lecture de valeurs de variables, depuis le clavier. Revenons à notre programme d'exemples,

notes

résumé

1m 37s



Lire une valeur au clavier

```
cin >> n;
```



et supposons que l'on ne veuille pas calculer que la valeur de 4 le carré de n , mais que l'on veuille pouvoir entrer depuis le clavier la valeur de n en la demandant à l'utilisateur. Comment ferait-on cela ? Eh bien, on va simplement insérer dans le programme, ici, de quoi demander à l'utilisateur d'entrer cette valeur, on commence bien sûr par lui afficher un message « Entrez une valeur pour n : » puis ensuite avec cette instruction, « `cin` redirigé vers n », on va pouvoir lire depuis le clavier la valeur de n . La lecture d'une variable au clavier se fait donc grâce à l'instruction `cin` (C - IN) `cin` veut dire le flot d'entrée standard, `in` comme input, c'est-à-dire que cela va représenter la lecture au clavier.

notes

résumé

2m 25s



Lire une valeur au clavier

cin représente le clavier

les différents éléments sont séparés par le symbole >>

nom de la variable dans laquelle sera stockée la valeur entrée au clavier

cin >> n;

Attention, uniquement des noms de variables peuvent figurer à droite du symbole >>

Par exemple, on ne peut pas faire:

~~cin >> "Entrez un nombre" >> n;~~

Par contre, on peut faire:

cin >> n1 >> n2 >> n3;

pour lire plusieurs valeurs à la suite.

Il faut faire:

```
cout << "Entrez un nombre" << endl;
cin >> n;
```

Puis on retrouve ici un signe similaire à celui qu'on avait utilisé pour l'affichage mais dirigé dans l'autre sens. L'information, ici, va du clavier vers votre variable n, si cela peut vous aider à vous souvenir de la façon d'écrire ce symbole. Et enfin, derrière, on va avoir le nom d'une variable dans lequel on va stocker la valeur qui sera lue au clavier. cin représente donc le clavier dans la mémoire, le programme, de l'ordinateur. Le symbole ici représente le fait que l'on aille de cin vers la mémoire, le sens de lecture, et puis, ensuite le nom de la variable. Attention, il faut bien comprendre que ce qui suit le signe de lecture ici, c'est bien le nom d'une variable. Ça ne peut pas être un message ou quoique ce soit d'autre. Ceci est absolument interdit. Si vous voulez afficher quelque chose, vous devez utiliser l'instruction cout. Vous commencez par afficher un message, éventuellement un retour à la ligne, c'est optionnel, puis ensuite, dans une seconde instruction, vous allez lire depuis le clavier dans la variable n. Le message s'affiche bien, non pas sur le clavier, mais sur le terminal cout et l'instruction de lecture est bien la ligne qui contient le cin ici. A signaler toutefois, que vous pouvez lire plusieurs variables à la suite, les unes derrière les autres, il faut simplement à chaque fois rajouter le signe de lecture. Ici, cela veut dire « depuis cin n1 puis lis n2 puis lis n3 ».

notes

résumé

3m 13s



Déroulement du programme pas-à-pas ^{EPFL}

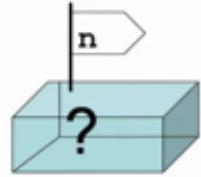
```
int n;
```

```
→ cout << "Entrez une valeur pour n:";  
cin >> n;
```

```
int n_carre;
```

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

```
cout << "La variable n contient " << n << "." << endl;
```



Ce qui s'affiche:

|

Personnellement, je vous encourage plutôt à l'écrire en plusieurs lignes séparées, donc écrire « lecture depuis cin dans n1 » puis « lecture depuis cin dans n2 » puis « lecture depuis cin dans n3 ». C'est peut-être un peu plus fastidieux, mais je crois que dans une première approche, c'est quand même beaucoup plus clair. Regardons maintenant notre exemple se dérouler pas à pas,

notes

résumé

4m 49s



Déroulement du programme pas-à-pas ^{EPFL}

```
int n;
```

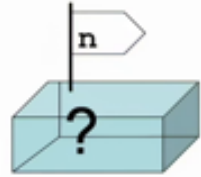
```
cout << "Entrez une valeur pour n:";
```

```
cin >> n;
```

```
int n_carre;
```

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

```
cout << "La variable n contient " << n << "." << endl;
```



Ce qui s'affiche:

```
Entrez une valeur pour n:
```

```
2
```

```
|
```

supposons que l'utilisateur entre la valeur 2 au clavier (il faut presser *return* une fois la valeur tapée)

on a donc déclaré au départ la variable `n`, puis on affiche la question, « Entrez une valeur pour n: » ce qui aura pour conséquences d'afficher ce message, puis on va lire au clavier. Supposons ici que l'utilisateur tape la valeur 2,

notes

résumé

5m 16s



Déroulement du programme pas-à-pas ^{EPFL}

```
int n;
```

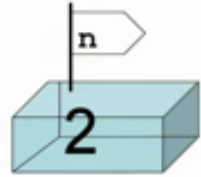
```
cout << "Entrez une valeur pour n:";
```

```
cin >> n;
```

```
int n_carre;
```

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

```
cout << "La variable n contient " << n << "." << endl;
```



Ce qui s'affiche:

```
Entrez une valeur pour n:
```

```
2
```

```
|
```

cette instruction cin dans n va faire que le 2 saisi au clavier va aller dans la variable n, ici.

notes

résumé

5m 37s



Déroulement du programme pas-à-pas ^{EPFL}

```
int n;
```

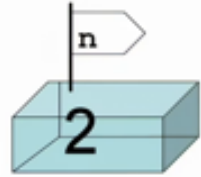
```
cout << "Entrez une valeur pour n:";
```

```
cin >> n;
```

```
int n_carre;
```

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

```
cout << "La variable n contient " << n << "." << endl;
```



Ce qui s'affiche:

```
Entrez une valeur pour n:
```

```
2
```

```
|
```

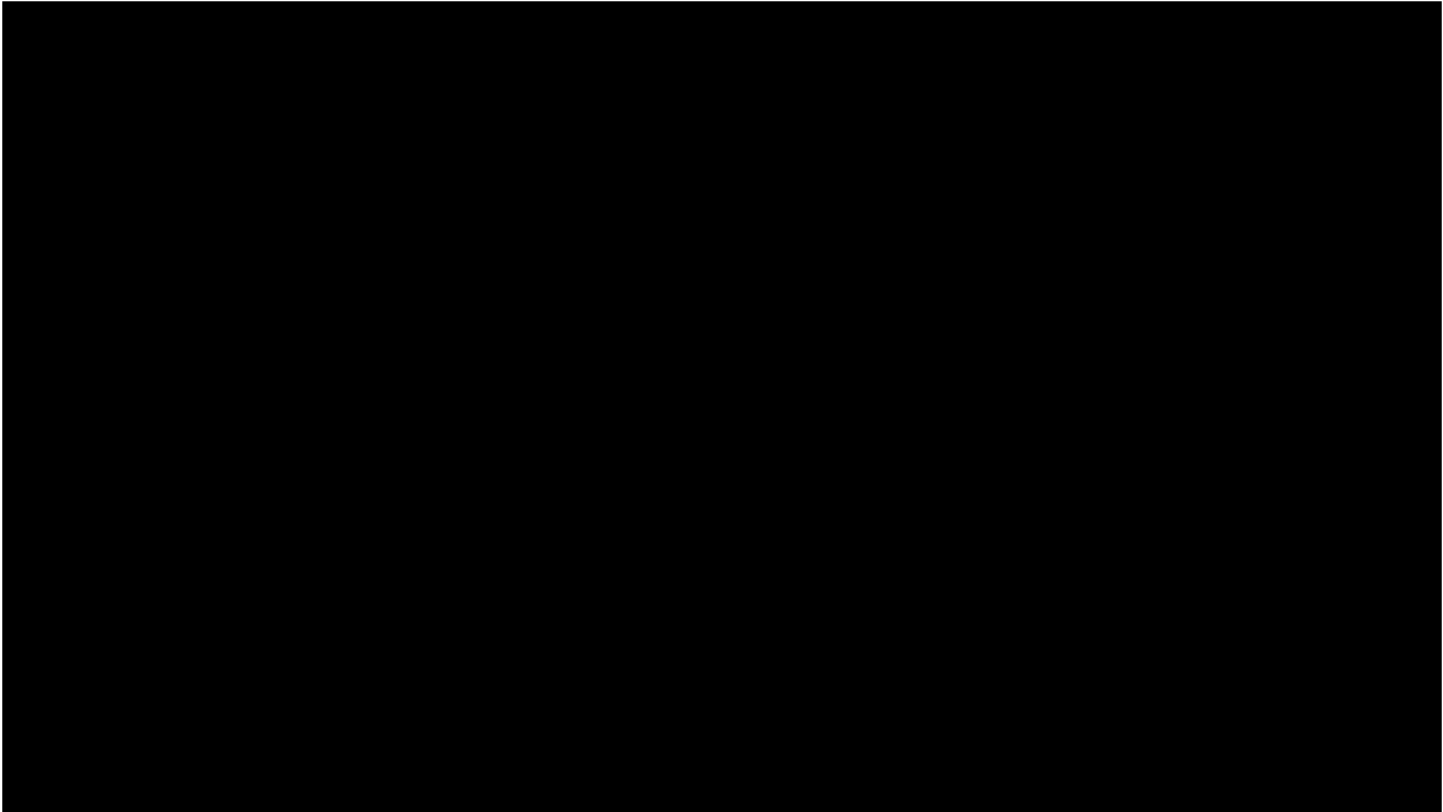
La variable n, une fois cette instruction exécutée à ce stade-là,

notes

résumé

5m 49s





contiendra la valeur qu'on a lue au clavier. Ensuite, la suite du programme se déroule exactement de la même façon que précédemment, On déclare par exemple la variable n^2 , ensuite on calcule le carré de n , n fois n , on l'affecte à la variable n^2 , puis on commence les affichages en commençant par l'affichage de « La variable n contient » - affichage de la valeur de n ; pour ce message, la variable n contient 2, et on continuerait le programme de la même façon que précédemment. de la même façon que précédemment.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

5m 50s



.....

.....

.....

.....

.....