

Support de cours

Cours:

Initiation à la programmation (en C++)

Vidéo:

Passage des arguments (partie 2)

Concepts (extraits des sous-titres générés automatiquement) :

Appel de fonction. Petit exemple introductif. Symbole particulier. En-tête de la fonction. Zone locale. Moment de cet appel. Petit programme principal. Moyen du symbole. Paramètre de la fonction. Variable locale. Valeur d'origine. Valeur. Altération. Val. Passage.



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



[vers la vidéo](#)

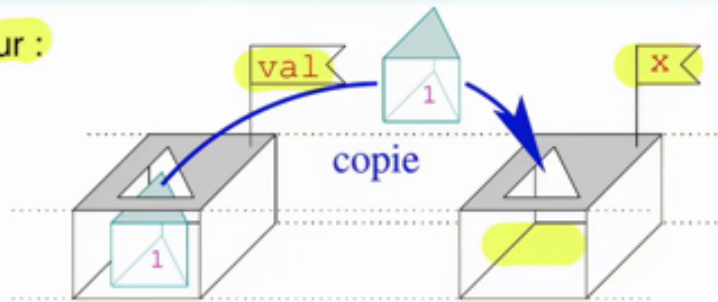
Center for Digital Education. Plus de matériel de soutien pédagogique ici :

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

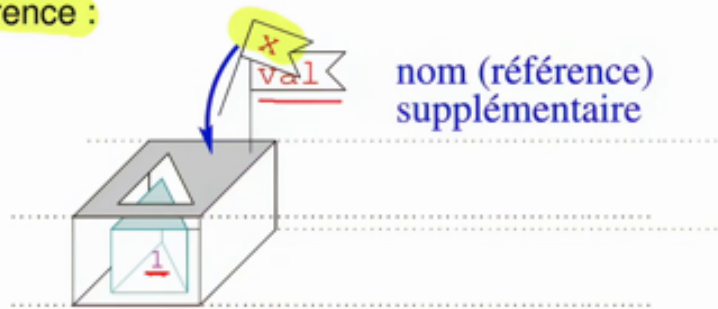
Passages d'argument : schéma



Passage par valeur :



Passage par référence :



...

notes

résumé

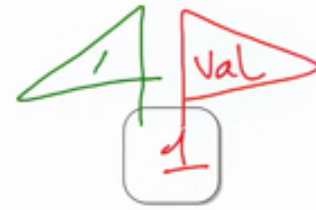
0m 0s



Exemple de passage par référence

```
void f(int& x) {
    x = x + 1;
    cout << "x=" << x;
}

int main() {
    int val(1);
    f(val);
    cout << " val=" << val << endl;
    return 0;
}
```



L'exécution de ce programme produit l'affichage :

x=2 val=2

Ce qui montre que les modifications effectuées à l'intérieur de la fonction **f()** **se répercutent** sur la variable extérieure **val** associée au paramètre **x** et passée par référence.

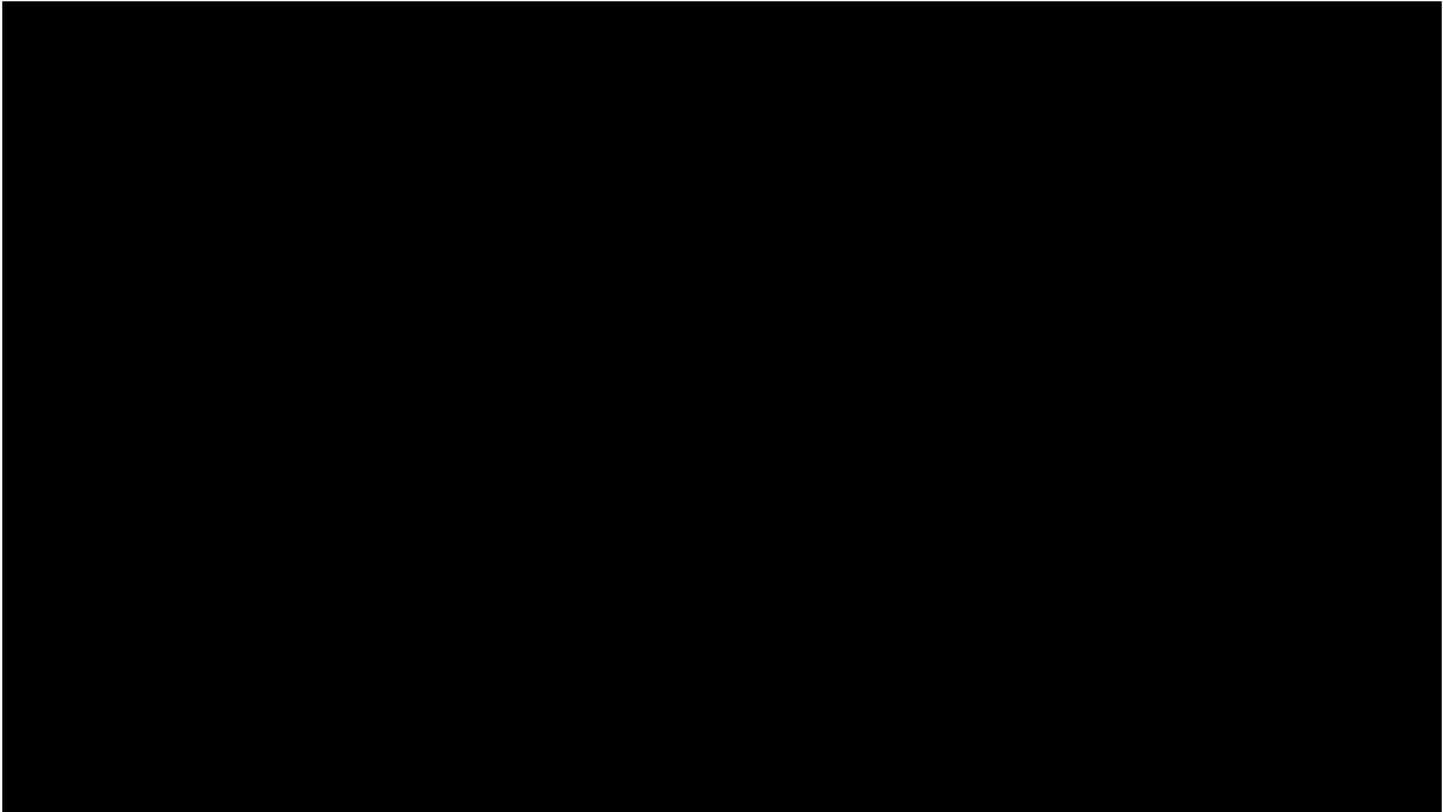
Si l'on prend maintenant notre petit exemple introductif, nous avons donc un petit programme principal qui déclarait une variable « val » donc le contenu est 1. Ensuite, il y avait un appel de fonction. Si l'on examine, l'en-tête de la fonction, il n'y a aucun symbole particulier, pas de « & », ce qui signifie que l'on utilise ici le passage par valeur. Puisqu'on utilise le passage par valeur, cela signifie que « x », le paramètre de la fonction est une zone locale et que donc la valeur de « val » est copiée à l'intérieur de « x ». L'altération de « x » n'a d'incidence que sur « x ». Si on affiche « x », on perçoit bien la modification de « x », par contre lorsqu'on a terminé d'exécuter la fonction et qu'on fait afficher la valeur de « val », on voit bien que la valeur d'origine reste inchangée, et donc que « val » vaut toujours 1. Si l'on reprend maintenant le même exemple mais qu'on indique que la fonction travaille par référence au moyen du symbole « & », nous avons toujours une variable locale au main qui contient 1, au moment de cet appel, vu qu'il y a un passage par référence, on indique simplement que « x » est un autre nom pour l'argument

notes

résumé

0m 1s





passé à la fonction, donc pour « val ». Ce qui veut dire ici qu'au moment où j'altère « x », j'altère également « val ». L'affichage de « x » me montre bien l'altération de « x », l'affichage de « val » me montre également l'altération de « val ». me montre également l'altération de « val ».

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

1m 25s



.....