

Support de cours

Cours:

Initiation à la programmation (en C++)

Vidéo:

Appels (partie 1)

Concepts (extraits des sous-titres générés automatiquement) :

Notion de fonction. Différentes étapes. Moyen d'un appel de fonction. Forme d'expression c. Instruction return. Première étape. Cas présent. Second argument. Situation concrète. Instruction terminale. Mot clé return. Troisième étape. Exemple du calcul des moyennes. Issue de la dernière étape. Fonction moyenne.



[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/>

Fonctions : appel

(Partie 1)

Initiation à la programmation (C++)

Vincent Lepetit, Jean-Cédric Chappelier et Jamila Sam

...

notes

résumé

0m 0s





Dans une séquence précédente, nous avons introduit la notion de fonction

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

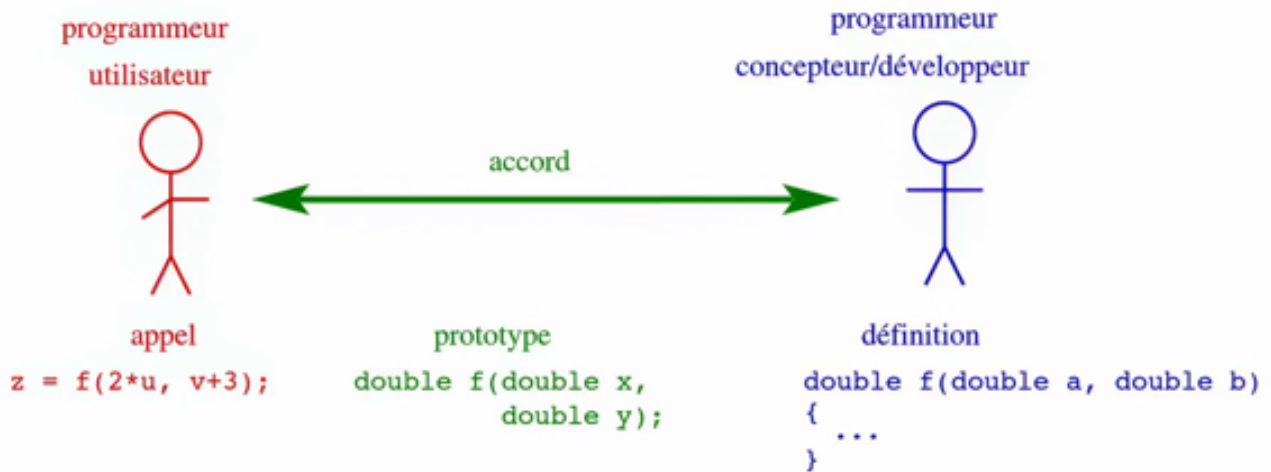
résumé

0m 1s



Les « 3 facettes » d'une fonction

- ▶ Résumé / Contrat (« prototype »)
- ▶ Création / Construction (« définition »)
- ▶ Utilisation (« appel »)



et son utilité en programmation. Nous allons aujourd'hui nous intéresser à ce qui se passe lorsqu'on utilise, lorsqu'on appelle une fonction. Concrètement, à ce stade, nous savons qu'à la notion de « fonction » sont associées 3 facettes : le prototype, la définition, l'appel,

notes

résumé

0m 5s



Exemple complet

```
#include <iostream>
using namespace std;

double moyenne(double nombre_1, double nombre_2);

int main()
{
    double note1(0.0), note2(0.0);
    cout << "Entrez vos deux notes : " << endl;
    cin >> note1 >> note2;
    cout << "Votre moyenne est : "
         << moyenne(note1, note2) << endl;
    return 0;
}

double moyenne(double x, double y)
{
    return (x + y) / 2.0;
}
```

et nous allons détailler ce qui se passe au moment où on appelle une fonction.

notes

résumé

0m 25s



Évaluation d'un appel de fonction

double moyenne (double x, double y)
{
 return (x + y) / 2.0;
}

Que se passe-t-il lors de l'appel suivant :

z = moyenne(1.5 + 0.8, 3.4 * 1.25);

1. $3.4 * 1.25 \rightarrow 4.25$
 $1.5 + 0.8 \rightarrow 2.3$

2. $x = 2.3$; $y = 4.25$

3. rien

4. $(x + y) / 2.0 \rightarrow 3.275$

5.

Reprenons l'exemple du calcul des moyennes. Nous avons ici le cas d'un petit programme principal qui demande à l'utilisateur d'introduire deux notes, qui lit ces deux notes depuis l'entrée standard et qui enfin affiche la moyenne de ces deux notes. Le calcul de la moyenne est réalisé au moyen d'un appel de fonction, tel que vous le voyez ici, et c'est les mécanismes appliqués par cet appel de fonction que nous allons maintenant étudier. Plaçons-nous dans une situation concrète où l'on souhaite calculer la moyenne de deux nombres, passer à la fonction moyenne sous la forme d'expression C++. Les arguments passés à la fonction moyenne au moment de l'appel correspondent évidemment aux paramètres attendus par la fonction pour qu'elle puisse s'exécuter. Notez que l'on appelle usuellement « paramètres » les données nécessaires à la fonction pour qu'elle puisse s'exécuter telle que décrite dans le prototype, et par « argument » les valeurs que l'on passe effectivement à la fonction au moment où on l'invoque, où on l'appelle. Donc, nous allons maintenant disséquer les différentes étapes qui ont lieu lorsqu'un tel appel est réalisé. Première étape : on commence par évaluer les arguments passés à la fonction moyenne au moment de l'appel. En C++, on ne peut pas présupposer de l'ordre selon lequel cette évaluation est faite, et donc ici on part de l'exemple de la situation où c'est le second argument qui est évalué en premier. Donc on évalue le second argument, ce qui a pour conséquence de produire la valeur 4.25. Ensuite, le premier argument est évalué à son tour, ce qui produit la valeur 2.3. Seconde étape : il s'agit maintenant d'apparier les arguments d'appel de la fonction avec les paramètres attendus par la fonction moyenne. Cet appariement se fait par le biais d'une affectation, donc on va affecter à x le résultat de l'évaluation du

notes

résumé

0m 29s



Évaluation d'un appel de fonction

double moyenne (double x, double y)
{
 return (x + y) / 2.0;
}

Que se passe-t-il lors de l'appel suivant :

z = moyenne(1.5 + 0.8, 3.4 * 1.25);

1. $3.4 * 1.25 \rightarrow 4.25$
 $1.5 + 0.8 \rightarrow 2.3$

2. $x = 2.3$; $y = 4.25$

3. rien

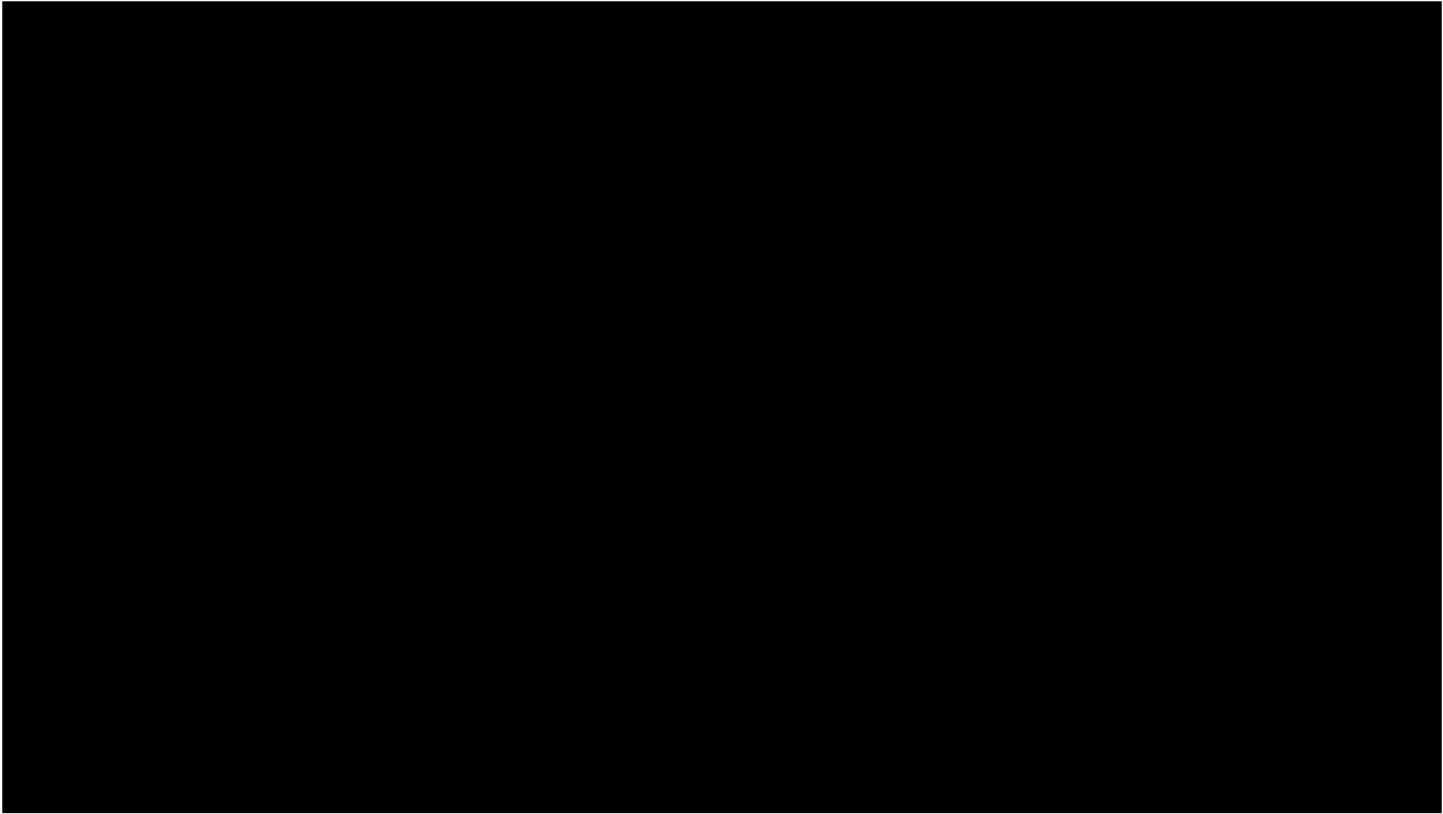
4. $(x + y) / 2.0 \rightarrow 3.275$

5.

premier argument, à savoir 2.3, et on va affecter à y le résultat de l'évaluation du second argument, à savoir 4.25. La fonction moyenne dispose désormais de valeurs concrètes dans x et y, avec lesquelles elle va pouvoir travailler, s'exécuter. Troisième étape : on commence l'exécution de la fonction moyenne. Ici, en l'occurrence, toutes les instructions précédant cette instruction terminale, l'instruction return vont être exécutées. Dans le cas présent, il n'existe aucune instruction précédant le return, donc la troisième étape ne fait rien de particulier. Enfin, l'instruction return est exécutée. On commence par évaluer l'expression suivant le mot clé return, donc ici, en l'occurrence, cette expression est évaluée avec des valeurs concrètes pour x et y, ce qui a pour résultat de produire la valeur 3.275.

notes

résumé



Ultime étape : on met la valeur produite par l'expression, on met cette valeur à disposition de celui qui a invoqué la fonction moyenne par le biais d'une instruction return, ce qui veut dire que cet appel de fonction peut désormais être remplacé par la valeur résultant de l'appel de fonction, à savoir 3.275. Donc, à l'issue de la dernière étape, la variable Z ôte ici la valeur 3.275. la variable Z ôte ici la valeur 3.275.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

3m 49s



.....

.....

.....

.....

.....