Support de cours

Cours:

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

Vidéo:

## Conditions (partie 3)

Concepts (extraits des sous-titres générés automatiquement) :

**Cas concret. Évaluation de cette expression. Opérateur logique. Cas d'un petit programme. Partie else du if. Opérateur logique usuel. Évaluation de l'expression. Partie positive du if. Conditions suivantes. Moyen du mot. Seule situation. Évaluation de cette seconde expression. Entrée standard. Nombre. Nombre conforme.**

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

vers la vidéo

EPFL

# Conditions
## (Partie 3)

Initiation à la programmation (C++)

Vincent Lepetit, Jean-Cédric Chappelier et Jamila Sam

...

notes

résumé

0m 0s

# Exemple avec l'opérateur logique and

```
cout << "Entrez un nombre entre 1 et 10:" << endl;
cin >> n;

if ((n >= 1) and (n <= 10)) {
  cout << "correct" << endl;
} else {
  cout << "incorrect" << endl;
}
```

Voyons maintenant sur un cas concret comment utiliser l'opérateur logique and. Nous avons ici le cas d'un petit programme qui demande à l'utilisateur d'introduire un

notes

résumé

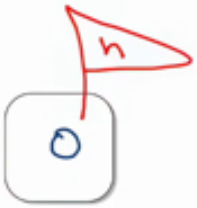
0m 1s

# Exemple avec l'opérateur logique and

```
cout << "Entrez un nombre entre 1 et 10:" << endl;
cin >> n;
if ((n >= 1) and (n <= 10)) {
  cout << "correct" << endl;
} else {
  cout << "incorrect" << endl;
}
```

false

false




nombre, et nous souhaitons que ce nombre soit compris entre deux bornes, 1 et 10. Le nombre est lu depuis l'entrée standard, et stocké dans une variable n. Nous voulons vérifier que le nombre introduit par l'utilisateur répond à nos attentes, est bien compris entre les bornes en question, et pour cela il faut que les deux conditions suivantes soient simultanément vérifiées. Il faut que le nombre introduit soit supérieur ou égal à 1 et inférieur ou égal à 10. Puisque l'on veut que les deux conditions soient simultanément vérifiées, il faut utiliser l'opérateur logique ET. Imaginons maintenant que l'utilisateur ait introduit la valeur zéro, ceci aura pour conséquence que l'évaluation de cette expression retournera "false". Une expression qui implique un and ne retourne "true" que si ces deux opérandes sont "true", ce qui n'est pas le cas ici. Donc le résultat ici de l'évaluation de l'expression avec le and est également "false". Par conséquent, nous allons brancher sur la partie else du if, et afficher "incorrect" pour signifier à l'utilisateur qu'il n'a pas

notes

résumé

0m 13s

# Exemple avec l'opérateur logique and

```
cout << "Entrez un nombre entre 1 et 10:" << endl;
cin >> n;
if ((n >= 1) and (n <= 10)) {
  cout << "correct" << endl;
} else {
  cout << "incorrect" << endl;
}
```



introduit un nombre conforme à nos attente. Si maintenant, par contre, l'utilisateur a introduit la valeur 5, l'évaluation de cette expression retournera "true", l'évaluation de cette seconde expression retournera également "true", ce qui aura pour conséquence que l'évaluation de l'expression avec le and retourne également "true".

notes

résumé

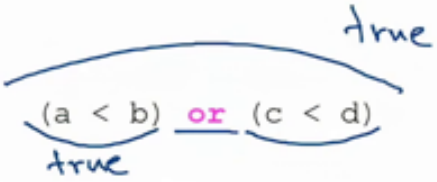
1m 13s

EPFL

# Les opérateurs logiques

**L'opérateur logique `or` (OU):**

par exemple, la condition




est vraie **si au moins une** des deux conditions `(a < b)` ou `(c < d)` est vraie.

**L'opérateur `or` peut aussi s'écrire `||` :** On aurait pu écrire

```
(a < b) || (c < d)
```

À ce moment-là nous allons brancher sur la partie positive du if et afficher "correct" pour signifier à l'utilisateur que le nombre introduit est bel et bien correct. Autre opérateur logique usuel, le OU, qui en C++ s'exprime soit au moyen du mot réservé or, soit au moyen de doubles barres verticales. Tout comme pour l'opérateur logique ET, l'opérateur logique OU requiert deux opérandes qui sont les expressions logiques, c'est-à-dire retournant "true" ou "false". L'évaluation de l'expression avec un OU retourne "true" si l'un ou l'autre, l'un ou l'autre, pas forcément les deux, l'un ou l'autre des opérandes est "true". On voit donc bien que la seule situation ou une expression logique impliquant un OU retourne "false" est une situation où les deux opérandes retournent "false". Dans ce cas-là, l'espression avec le or retourne aussi "false". retourne aussi "false".

notes

résumé

1m 37s