

Information, Calcul, Communication (partie programmation) : Entrées / Sorties

Jean-Cédric Chappelier

Faculté I&C

Rappel du calendrier

MOOC		décalage / MOOC	exercices prog. 1h45 Jeudi 8-10	cours prog. 45 min. Jeudi 10-11
1	10.09.26 --	-1	prise en main	Bienvenue/Introduction
2	17.09.26 1. variables	0	variables / expressions	variables / expressions
3	24.09.26 2. if	0	if – switch	if – switch
4	01.10.26 3. for/while	0	for / while	for / while
5	08.10.26 4. fonctions	0	fonctions (1)	fonctions (1)
6	15.10.26	1	fonctions (2)	fonctions (2)
-	22.10.26			
7	29.10.26 5. tableaux (vector)	1	vector	vector
8	05.11.26 6. string + struct	1	array / string	array / string
9	12.11.26	2	structures	structures
10	19.11.26 7. pointeurs	2	pointeurs	pointeurs
11	26.11.26	-	entrées/sorties	entrées/sorties
12	03.12.26	-	erreurs / exceptions	erreurs / exceptions
13	10.12.26	-	révisions	théorie : sécurité
14	17.12.26 8. étude de cas	-	révisions	Révisions

Objectifs du cours d'aujourd'hui

- ▶ Rappels sur les entrées/sorties
- ▶ Etude de cas

std::cin

- ▶ >> s'arrête juste avant le premier blanc rencontré
 - ▶ utiliser `getline(cin, chaine)` si nécessaire
 - ▶ penser à le faire précéder de `cin >> ws;`
- ▶ comprendre la différence entre `cin.fail()`, `cin.clear()` et `cin.ignore()`

Schéma général d'utilisation des flots

- **création d'un flot** : **déclaration** d'une variable de `ifstream` ou `ofstream`

```
ifstream flot;
```

- **lien** de la variable déclarée avec un fichier

```
flot.open("nom");
```

```
ifstream flot("nom");
```



+ **tester la validité** : `not flot.fail()`

- **utilisation** de la variable déclarée et liée

- s'utilise comme `cin` ou `cout`

- dans un contexte de test, les opérateurs `<<` et `>>`, ainsi que `getline()`, retournent
« `not flot.fail()` »,

- tester la fin du fichier avec `flot.eof()`.

- **fermeture** du flot

```
flot.close();
```

Entrées/Sorties formatées

On peut formater ses entrées/sorties :

- ▶ soit des **manipulateurs** appliqués à l'instruction `<<`
(`#include <iomanip>`)
`cout << manipulateur << expression << ...`
- ▶ soit des **options de configurations** pour une variable de type `ostream`
`flot.setf(ios::option);`
`flot.unsetf(ios::option);`

Etude de cas

Dans un fichier texte dont on demande le nom à l'utilisateur, dessiner un « sapin » (par exemple :

```

~
|
*
***
*****
***
*****
*****
*****
*****0**
***
*****
*****
*****0**
*****
*****0***0
*****
*****0*****0**
#####
#####
#####

```