

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

Jean-Cédric Chappelier

Laboratoire d'Intelligence Artificielle  
Faculté I&C

# Rappel du calendrier

	MOOC	décalage / MOOC	exercices prog.	cours prog.
			1h45	45 min.
			Jeudi 8-10	Jeudi 10-11
1	11.09.25	--	-1 prise en main	Bienvenue/Introduction
2	18.09.25	1. variables	0 variables / expressions	variables / expressions
3	25.09.25	2. if	0 if – switch	if – switch
4	02.10.25	3. for/while	0 for / while	for / while
5	09.10.25	4. fonctions	0 fonctions (1)	fonctions (1)
6	16.10.25		1 fonctions (2)	fonctions (2)
-	23.10.25			
7	30.10.25	5. tableaux (vector)	1 vector	vector
8	06.11.25	6. string + struct	1 array / string	array / string
9	13.11.25		2 structures	structures
10	20.11.25	7. pointeurs	2 pointeurs	pointeurs
11	27.11.25		- entrées/sorties	entrées/sorties
12	04.12.25		- erreurs / exceptions	erreurs / exceptions
13	11.12.25		- révisions	théorie : sécurité
14	18.12.25	8. étude de cas	- révisions	Révisions
				(ne sont pas sur le MOOC)

# 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 précédé 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**  
`fout.setf(ios::option);`  
`fout.unsetf(ios::option);`

# Etude de cas

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

```
^  
|  
*  
***  
*****  
***  
*****  
*****  
*****  
*****  
*****  
***  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
#  
#  
#  
#
```