

# 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. 1h45 Jeudi 8-10	cours prog. 45 min. Jeudi 10-11
1	12.09.24 --	-1	prise en main	Bienvenue/Introduction
2	19.09.24 1. variables	0	variables / expressions	variables / expressions
3	26.09.24 2. if	0	if – switch	if – switch
4	03.10.24 3. for/while	0	for / while	for / while
5	10.10.24 4. fonctions	0	fonctions (1)	fonctions (1)
6	17.10.24	1	fonctions (2)	fonctions (2)
-	24.10.24			
7	31.10.24 5. tableaux (vector)	1	vector	vector
8	07.11.24 6. string + struct	1	array / string	array / string
9	14.11.24	2	structures	structures
10	21.11.24 7. pointeurs	2	pointeurs	pointeurs
11	28.11.24	-	entrées/sorties	entrées/sorties
12	05.12.24	-	erreurs / exceptions	erreurs / exceptions
13	12.12.24	-	révisions	théorie : sécurité
14	19.12.24 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);`

