

# Programmation Orientée Objet : Templates



Jean-Cédric Chappelier

Laboratoire d'Intelligence Artificielle  
Faculté I&C

# Organisation du travail (semestre)

	MOOC	déc.	cours 1 h Jeudi 8-9	exercices 2 h Jeudi 9-11
1	20.02.25		0	Intro + compil. séparée
2	27.02.25	1. Intro POO	0	Intro POO
3	06.03.25	2. Constructeurs/Desi	0	Constructeurs
4	13.03.25	3. Surcharge des opé	0	Surcharge
5	20.03.25	4. Héritage	0	Héritage
6	27.03.25	5. Polymorphisme	0	Polymorphisme 1
7	03.04.25		1	Polymorphisme 2 / Collections hétérogènes
8	10.04.25	6. Héritage multiple	1	Héritage multiple
9	17.04.25		-	Midtem
-	24.04.25		-	vacances Pâques
10	01.05.25	(7. Etude de cas)	-	Templates
11	08.05.25		-	Structure de données abstraites ; Bibliothèques
12	15.05.25	(7. Etude de cas)	-	Bibliothèques (fin) + Révisions
13	22.05.25		-	Examen
14	29.05.25		-	(Ascension)

# Objectifs de la leçon d'aujourd'hui

- ▶ Concepts fondamentaux
- ▶ Étude de cas

# Concepts fondamentaux

- ▶ bien comprendre de quoi il s'agit :  
modèles de programmes  
(« infinité potentielle de codes »)
- ▶ modèles dans le(s) type(s) et/ou dans des entiers
- ▶ instanciations implicites et explicites
- ▶ spécialisation de modèles

class  
fct<sup>o</sup>

template < type non T >  
template < int >

code

template <...>

A code 1

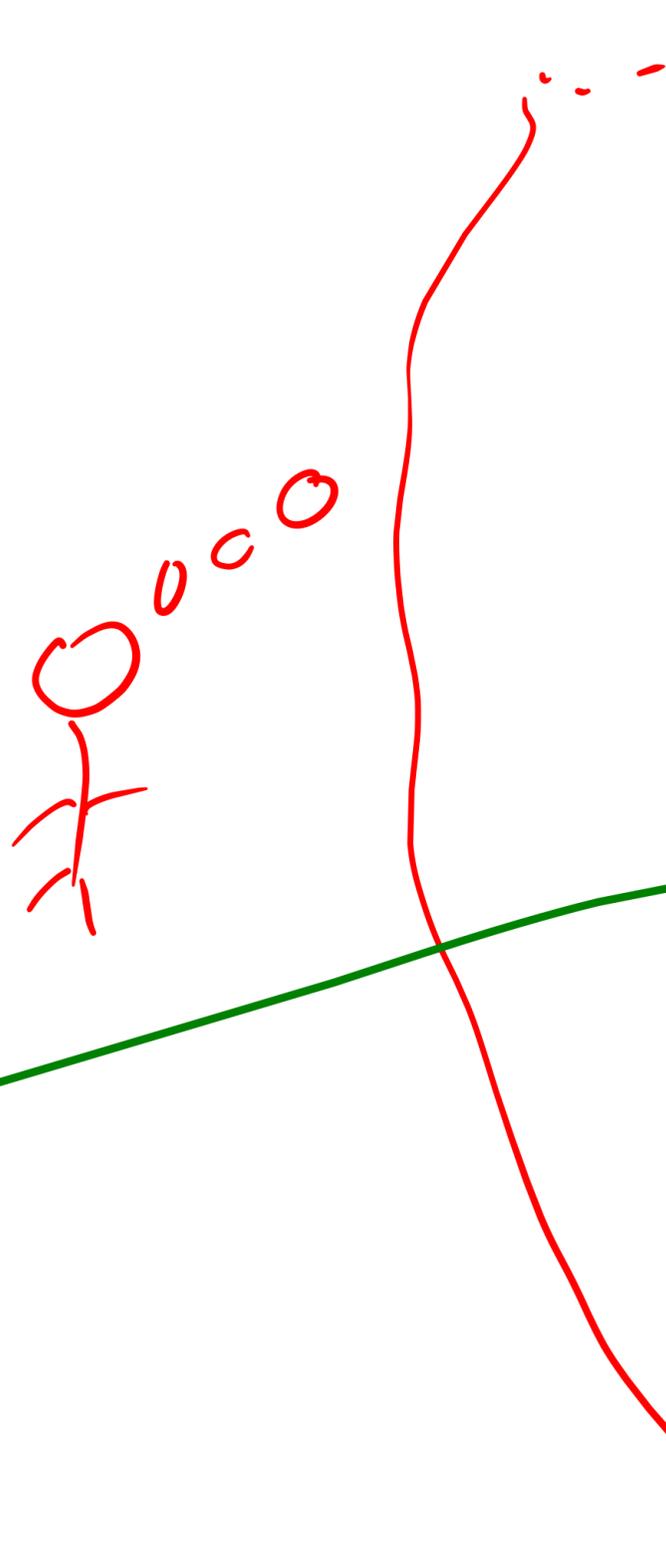
instantiation

A <int>

(implicit)

Specialisation

template <> A <double>  
code 2



A code 1

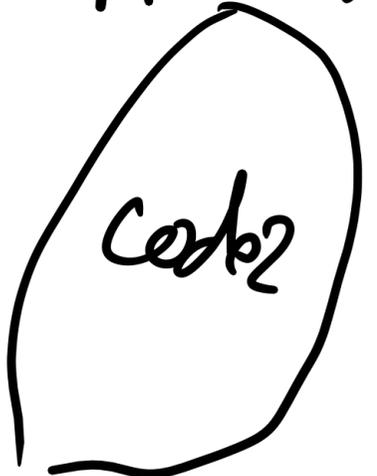
A code 1

A <int> code 1

A code 1

A <double>

A code 1



# Etude de cas

- ▶ vecteurs (mathématiques) de dimension finie sur un corps :

☞ template en :

- ▶ la dimension
- ▶ le corps, double par défaut

→ "int"      size\_t

↳ typename

Intérêts :

- ▶ template de classe
- ▶ template de fonctions (surcharges externes)
- ▶ template méthode dans le template de classe  
(convert()) pour changer de dimension)
- ▶ problème de conversion/instanciation implicite lorsqu'ambigüe

$V_3 \leftarrow \mathcal{V}_2, \text{convert}()$

