
Intitulé de l'Unité d'Enseignement (UE) : Programmation Objet C++**Responsable de l'UE :** Alice Cohen-Hadria (alice.cohenhadria@ircam.fr)**Code UE :** UM4RBI22**Volumes horaires / étu :** 8h Cours, 4h TD, 12h TP**Nombre de crédits de l'UE/EC :** 3 ECTS

Parcours-type :	- E3A :	☐ CIMES	☐ Syscom	☐ IPS
	- AR :	☒ ROBOT	☒ SI	☒ MeDH
	- MECA :	☐ MS2	☐ MF2A	☐ CompMech
		☐ ACOU	☐ EE	☐ EE APP

Semestre où l'enseignement est proposé : ☐ S1 ☒ S2 ☐ S3 ☐ S4**Langue d'enseignement :** ☒ Français ☐ Anglais**Public concerné :** ☒ Département SDI ☐ Autre (préciser) :**Localisation :** ☒ Campus PMC ☐ Autre (préciser) :

Thématique

Cette UE forme les étudiantes et étudiants à la programmation objet avancée en C++. Les fondamentaux de l'héritage, de l'encapsulation et du polymorphisme d'objet sont abordés dans un premier temps. Dans un second temps, différents patrons de conceptions seront abordés.

Mots-clés : Programmation orientée objet, héritage, polymorphisme, encapsulation, template, patron de conception

Prérequis

Afin de suivre cette UE, doivent être acquis précédemment :

- Ecrire un programme simple en utilisant des variables, structures de contrôles à l'intérieur de classes et de méthodes.
- Tester ses fonctions et classes avec différents scénarios de tests décrits
- Passer d'un diagramme UML simple (une seule classe) au programme correspondant, et inversement

Acquis d'apprentissage visés (AAV)

À l'issue de cette UE, les étudiantes ou étudiants seront capables de :

1. Créer un programme de test avec une description des tests réalisés et de leurs résultats
2. Créer et compiler un makefile
3. Savoir se servir de shell (se déplacer, créer, copier, lister, supprimer, chercher, droits d'utilisateur)
4. Ecrire la documentation d'une fonction (/** */) et générer le fichier doxygen correspondant
5. Utiliser un code précompilé avec la documentation
6. Passer d'un énoncé décrivant une situation à un diagramme de classes UML, puis au code correspondant (et inversement)
7. Factoriser un code (créer des classes mères à partir de classes existantes)
8. Savoir implémenter un patron de conception à partir d'un énoncé
9. Savoir mettre en oeuvre les mécanismes de l'héritage et du polymorphisme associé
10. Savoir utiliser les templates

Séquencement et modalités de l'enseignement (indicatif)

Semaine	C/TD/TP	Contenu	Préparation	AAV associé(s)
S1	C1 /TP1 (2h)	Introduction C++ et généralités : compilation, makefile, shell, documentation		AAV1, 2, 3, 4
S2	C2	Héritage, Polymorphisme, Encapsulation		AAV9, 6, 7, 1, 4
S3	TD1 /TP2 (2h)	sur C2		
S4	C3	Template, librairies		AAV 10, 5, 1, 4
S5	TD2/TP3 (2h)	Sur C3		
S6	TP Solo 4 (2h)	Porte sur TP1 et 2 : quizz 30min, TP solo 30 min		
S7	C4	Patron de conception		AAV8, 1, 4
S8	TP5 (4h)	Sur C4 + préparation projet		
S12	TP6	Soutenance de projet		
S15	Examen réparti	Examen final écrit sur l'ensemble de l'UE		

Déroulement de l'UE

Le premier TP suit directement le C1 et porte sur la prise en main de makefile et de shell.

Ensuite, après chaque CM, le TD permet de mettre en pratique les notions vues en cours avec des exercices simples, les TPs permettent aux étudiantes et étudiants de coder en autonomie.

Modalités d'évaluation des acquis d'apprentissage (1ère session)

Semaine	Indiv/collectif	Présentiel/distanciel	Type d'examen	AAV évalué(s)	Barème %
S6	Individuel	Présentiel	TP	1,2,4,6,7,9	20 %
S12	Collectif	Présentiel	Autre	8,9,10,1,2	30 %
S15	Individuel	Présentiel	Écrit	1,2,3,4,5,6,7,8,9,10	50 %

2nde session

Session	Indiv/collectif	Présentiel/distanciel	Type d'examen	AAV évalué(s)	Barème %
2	Individuel	Présentiel	Écrit		50 %
1	Individuel	Présentiel	TP	1,2,4,6,7,9	20 %
1	Individuel	Présentiel	Autre	8,9,10,1,2	30 %