

Programmation orientée objets

Domaine	Ingénierie et Architecture
Filière	Génie électrique
Orientation	Electronique et Automatisation industrielle (EAI)
Mode de formation	Plein temps

Informations générales

Nom:	:	Programmation orientée objets
Identifiant:	:	ProgOO
Années académiques	:	2018-2019, 2019-2020, 2020-2021
Responsable:	:	Yves Chevallier
Charge de travail:	:	150 heures d'études
Périodes encadrées:	:	96 (= 72 heures)

Semestre	E1	S1	S2	E2	S3	S4	E3	S5	S6
Cours					32				
Laboratoire					64				

Connaissances préalables recommandées

L'étudiant doit connaître et savoir utiliser les notions suivantes : programmation structurée en langage C. Les unités d'enseignement INFO1 et INFO2 (analyse et programmation) permettent d'acquérir ces connaissances.

Objectifs

A l'issue de cette unité d'enseignement, l'étudiant sera capable de :

- expliquer les éléments de base apportés par le langage C++ par rapport au langage C.
- expliquer les concepts POO (programmation orientée objet) suivants : objet, classe, attribut, méthode, encapsulation, constructeur, interface, classe abstraite, propriété statique, héritage, polymorphisme, redéfinition et liaison dynamique, copie superficielle/profonde, exceptions et généricité ;
- utiliser les concepts POO du langage C++ et énoncer le vocabulaire associé aux spécificités de ce langage ;
- expliquer et utiliser les spécificités du langage C++ : destructeur, espace de nommage, approche de l'encapsulation, constructeur de copie, héritage simple et multiple, méthode et classe virtuelles, fonction, méthode et classe amies et surcharge des opérateurs ;
- Développer un programme combinant des éléments en C et en C++ ;
- Expliquer le fonctionnement interne de la liaison dynamique dans le cas d'un héritage simple, et expliciter son impact sur les performances pour des applications temps-réel ;
- Mettre en oeuvre quelques conteneurs et algorithmes de la STL;

- Modéliser la structure d'un programme au moyen d'un diagramme de classes (UML) au niveau de la conception et de la documentation ;

- Citer et expliquer quelques design patterns GOF et MVC, identifier des situations appropriées pour les mettre en oeuvre, et les implémenter en C++.

A l'issue des travaux de laboratoire, l'étudiant sera en outre capable de :

- Appliquer les principes de la POO pour réaliser des applications présentant un domaine métier polymorphe;

- Concevoir une couche d'abstraction avec les mécanismes de la POO pour garantir l'indépendance à un matériel/logiciel.

- Mettre en oeuvre la POO dans un contexte présentant des contraintes temps-réel, comme la commande de machines.

Contenu et formes d'enseignement

Répartition des périodes indiquée à titre informatif.

Cours: 32 périodes

- Introduction.	1
- De C à C++ : les apports du C++ à la programmation structurée.	3
- Rappel sur les notions de TDA. Exercices d'identification d'un TDA.	2
- La programmation orientée objet en C++	6
- Conception de modèles de classes avec UML	2
- La programmation générique en C++	6
- La bibliothèque STL	2
- Modèles de conception réutilisables : MVC, GOF	4
- Approche des tests unitaires avec C++	2
- Contrôle continu	4

Laboratoire: 64 périodes

- Prise en main du C++	4
- Mise en oeuvre de la programmation orientée objet	20
- Mise en oeuvre de la programmation générique	10
- Réalisation de tests unitaires	4
- Programmation graphique	6
- Mini-Projet - Conception et développement	20

Bibliographie

Aucune information

Contrôle de connaissances**Cours:**

l'acquisition des matières de cet enseignement sera contrôlée au fur et à mesure par des tests et des travaux personnels tout au long de son déroulement. Il y aura au moins 3 tests d'une durée totale d'au moins 2 périodes.

Laboratoire:

ils seront évalués sur la base des rapports de manipulation, à 3 reprises au minimum.

Calcul de la note finale

Note finale = moyenne cours x 0.5 + moyenne laboratoire x 0.5

Fiche validée le 21.08.2018 par Bossoney Luc