

Programmation Orientée Objet

Mise en oeuvre en C++

0 - Détails et organisation

Julien Yves ROLLAND (julien.rolland@univ-fcomte.fr)

Laboratoire de Mathématiques de Besançon
Université de Franche-Comté

0.1

Présentation

Présentation de l'intervenant

Julien Yves Rolland

- Ingénieur de Recherche au laboratoire de Mathématiques
- Docteur en Physique des transferts
- Bureau 313B (prévenir par mail ¹ la veille)

Expérience

- Utilisation du C++ et Python depuis ma thèse (2006)
- Production de code pour le calcul scientifique
- 5 ans d'enseignement en école d'ingénieur(Mines et réseau polytech)
- 6 ans d'enseignement en universités(Nantes et Besançon)

0.2

Déroulement

Planning

Planning « provisoire » des séances

Créneaux horaires habituels

- Mardi matin de 9h30 à 12h30
- Jeudi après-midi de 13h30 à 16h30
- Sauf semaines 41 et 46

Répartitions des heures

39h d'enseignements réparties :

- 12 séances en 320B (cours, TD, TP)
- une séance de mini-projet évalué de 3h en 320B

Semaine	37	38	39	40	41	42	43	44	45	46
Mardi 9h30	320B	320B	320B		320B	320B			320B	
Mardi 13h30										320B
Jeudi 9h30					320B					
Jeudi 13h30	320B		320B	320B		320B				320B

1. julien.rolland@univ-fcomte.fr

0.3

Évaluation

Évaluation de l'enseignement

Modalité d'évaluation

- 1 séance de mini-projet de 3h en salle 320B (18/11)
- 1 examen écrit de 2h (lieu et date à définir)

Pondération

- UE « Programmation Orientée Objets » de 6 ECTS
- Partagée à 50% avec le « Projet C++ » (J-J. Duchamps)
- Examen représente 60% de la note, le mini-projet 40%
- La présence est obligatoire en cours, TD et TP. Les absences non justifiées seront pénalisées.

0.4

Objectifs

Objectifs du module

- Bases du langage C++
 1. Fonctionnement et structure d'un programme
 2. Types et opérations de base
 3. Structures de contrôle
 4. Programmation procédurale
- Abstraction en C++ : Les classes
 1. Types définis par l'utilisateur
 2. Concepts et outils de l'abstraction
 3. Notions fondamentales de la POO
 4. Mise en pratique concrète des classes
 - Encapsulation
 - Héritage
 - Polymorphisme
- Programmation générique et cas concrets
 1. Notion de programmation générique (Template)
 2. Présentation et usages de la STL
 3. Détails de cas concrets

0.5

Supports et références

Matériels

Matériels du module

Support du cours

L'intégralité du cours sera disponible en téléchargement ².

- Support de présentation
- Support de cours version imprimable
- Principaux exemples de codes

Modalité pour l'examen

- Support version imprimable autorisé pour le mini-projet
- Aucun document pour l'examen écrit

Machines

- TP sur machines Windows en salle 320B
- 6 machines disponibles hors cours en salle 304B
- GCC 7+, IDE Code : :Blocks 20.03

0.6

2. <http://lmb.univ-fcomte.fr/Master-1-POO> ou <http://lmb.prod.lamp.cnrs.fr/Master-1-POO>

Convention de notations

Informations

Ce bloc indique en grande majorité une définition ou un concept à connaître.

Information complémentaire

Ce bloc indique une information complémentaire dont la connaissance est parfois bien utile.

En pratique

Ce bloc indique comment est utilisé un concept concrètement dans les cas les plus classiques.

Le reste du texte donne un complément d'information ou un exemple.

0.7

Références

Quelques références

Ouvrages

- « The C++ Programming Language » Fourth Edition *Bjarne Stroustrup* – Addison-Wesley
- « Programming Principles and Practice Using C++ » *Bjarne Stroustrup* – Addison-Wesley
- « Effective Modern C++ » *Scott Meyers* – O'Reilly

Sites Internet

- <http://www.stroustrup.com/>
- <http://www.cplusplus.com/> *Section Tutorial - « C++ Language »*
- <http://fr.cppreference.com/> *Section « Référence C++ » (anglais dispo)*

0.8