

IDENTIFICATION	
Intitulé de l'Unité de formation (d'enseignement) : <b>POO D'APPLICATIONS TECHNIQUES</b>	Niveau d'études supérieures : <b>BAC.INF. ET SYSTÈMES</b>
Intitulé du cours : <b>Programmation orientée objet d'applications techniques</b>	Nombre de crédits ECTS : <b>10</b>
Nombre de périodes : <b>100</b>	Code : N°UE <b>940</b>

DESCRIPTION
<p>Prérequis : <a href="http://admin.segec.be/documents/7041.pdf">http://admin.segec.be/documents/7041.pdf</a> (pg.53, 54 : capacités préalables requises)</p> <p>En informatique appliquée aux sciences et aux technologies : bases de la programmation, effectuer l'analyse, la programmation et le test d'une ou plusieurs applications simples en mode console faisant intervenir des boucles em boîtées, des tableaux à 2 dimensions, la conception et l'emploi de procédures ou de fonctions, y compris le choix de la portée des variables.</p> <p>En analyse informatique, à partir d'un cas réel et concret,</p> <ul style="list-style-type: none"> <li>❖ mettre en œuvre une stratégie cohérente de résolution de problème en faisant preuve d'appropriation des concepts méthodologiques et technologiques à chaque étape du cycle de vie du projet informatique ;</li> <li>❖ choisir, de construire et de représenter le(s) modèle(s) correspondant(s) en utilisant une notation adaptée ;</li> <li>❖ respecter les règles de modélisation et les spécifications du problème ;</li> <li>❖ justifier la démarche et les choix mis en œuvre.</li> </ul> <p>En initiation aux bases de données, à partir d'un cahier des charges, en disposant d'une station informatique opérationnelle équipée d'un logiciel « Bases de données », développer et gérer une base de données sur un système de gestion de bases de données relationnelles et de manipuler des requêtes sous un langage tel que SQL,... dans des cas simples.</p> <p>Avoir réussi les différentes UE sujettes à ces prérequis.</p>
<p><del>Documents de référence pour une préparation préalable au cours.</del> Avoir revu (et assimilé) les prérequis. Il est très important d'avoir révisé avant d'aborder ce cours !</p> <p>L'essentiel de ces EU ayant été assimilée en première et deuxième années (de la section), il a été conseillé aux étudiants (<i>suivant les chargés de cours de l'époque</i>) d'entretenir ces acquis étant donné que cette dernière UE940 ne se donnent (<i>malheureusement il est vrai</i>) qu'en quatrième année (dans cette section) ! Ce qui laisse donc parfois (<i>au minimum</i>) plus d'une année calendrier aux étudiants sans encore pratiquer les prérequis à cette UE lors de leur formation académique. Les principales lacunes constatées étant bien souvent le manque de pratique entre temps surtout des langages de programmation (et une pénible remise à niveau) ! Ainsi donc, le (re)démarrage du cours UE940 tiendra compte « autant que faire ce peut » de cette longue période sans cours de programmation. Mais, ceci dit, certainement pas pour autant à un niveau de (re)démarrage à « zéro » en programmation (<i>ni même en : analyse, base de données, réseau, ...</i>) !</p>

Objectifs : <http://admin.segec.be/documents/7041.pdf> (pg.53 : finalités)

Cette unité de formation vise à permettre à l'étudiant :

- ❖ d'effectuer l'analyse informatique, la programmation orientée objet et le test d'une application technique avec interface graphique ;
- ❖ de développer des compétences relationnelles dans le cadre d'un travail d'équipe.

Il s'agira donc bien de s'assurer que les étudiants soient capables au final de développer des « applications » (orientées objets) en diffusant l'information (cette fois) dans des interfaces graphiques (plus précisément des IHM – interfaces hommes-machines « graphiques »). Le tout en utilisant des techniques de développement pour pouvoir aussi s'intégrer dans des développements collaboratifs plus larges.

Contenu du cours : <http://admin.segec.be/documents/7041.pdf> (pg.54 : programme)

Dans le cadre d'applications relevant du domaine technique, l'étudiant sera capable d'effectuer :

- ❖ l'analyse informatique par des méthodes peu ou non formalisées,
- ❖ la programmation dans un langage orienté objet,
- ❖ le test d'applications du domaine des techniques et des sciences :
  - en employant correctement un environnement de développement intégré,
  - en utilisant des concepts de base de la programmation orientée objet (encapsulation, héritage, polymorphisme,...),
  - en utilisant, de manière élémentaire, l'algorithmique scientifique classique (analyse numérique, emploi d'un générateur de nombres pseudo-aléatoires, régressions,...),
  - en exploitant une bibliothèque de classes pour l'accès à des bases de données relationnelles locales,
  - en exploitant une bibliothèque de classes pour l'interface graphiques et la création de graphiques,
  - en structurant et en documentant les programmes.

Les points exposés de ce programme sont « clairs » et seront « traités » au rythme des cours en classe. Toutefois, nombre de ces points présupposent (pour rappel) que les acquis doivent être suffisants dans les UE ayant précédé, pour poursuivre avec ce cours.

*Voir ci-dessous, concernant la « méthodologie » du chargé de cours pour l'étude de ce programme...*

#### Bibliographie

Tous les supports de cours sont rédigés (composés, démythifiés, ...) par le chargé de cours. Ces supports sont envoyés aux étudiants par courriel, directement par le chargé de cours, au rythme des cours et des avancées du groupe-classe. Ils sont exposés en classe et servent de support durant les cours donc, mais aussi pour le travail d'étude des étudiants sur les différents points de la matière enseignée. Ce n'est que si nécessaire, que certaines références « constructeurs ou certifiées » sont citées en plus ou référencées dans les supports des cours, pour permettre aux étudiants de poursuivre au-delà du cours.

Statut du cours : **unité déterminante !**

### PERSONNEL ENSEIGNANT

Chargé de cours : M. **Stéphane Ghidetti** (depuis janvier 2014 à l'IRAM), informaticien. Chargé « aujourd'hui » de différents cours (au rythme des attributions suivant les années académiques) à

l'IRAM en : Bachelier en Informatique de Gestion, Bachelier en Informatique et Systèmes, Brevet Webdesigner. Faisant suite ainsi préalablement à 20 années d'expérience professionnelle en Belgique et à l'international (EMEA) dans l'analyse et le développement (*de solutions logiciels C/S, de solutions applicatives Internet et de sites Web, de progiciels SCM/SRM/CRM*) comme (*en bref :*) Architecte d'applications Internet ; Directeur de projets « Software engineering » ; ... et, « *depuis janvier 2014* », ayant décidé de consacrer son expérience, mais aussi « en parallèle » la suite de son « étude des sciences de l'informatique », au secteur de la formation, avec comme public cible les étudiants futurs acteurs dans l'« informatique ». Ce choix, de participer et de contribuer à la formation, m'étant personnellement possible (dans ces sections informatiques) qu'en étant aujourd'hui encore et toujours en relation (suite à l'expérience professionnelle passée) avec les « personnes *ad hoc* à l'international » qui : d'une part, continuent « d'inventer, de concevoir, de préparer, ... » les technologies informatiques d'aujourd'hui « et » de demain ; et d'autre part, mettent en oeuvre ces technologies dans des projets « opérationnels ».

## MÉTHODOLOGIE

Afin d'aider les étudiants à répondre aux exigences de cette « unité d'enseignement du Dossier pédagogique » (de sa finalité, son programme et des capacités terminales), le cours « en fonction du calendrier de cette année scolaire » sera articulé en deux temps, à savoir :

- durant les cours en soirées, tous les concepts de « base de l'orienté objets » du programme seront abordés (et le respect des bonnes pratiques « orientées objets » du langage associé) ;
- durant les cours du samedi, seront exploités plus spécifiquement ces concepts de base de l'orienté objets directement dans l'implémentation des interfaces graphiques (IHM).

Le cours sera donc exposé suivant un rythme de présentations continues (dans la partie plus « orientée objets » et dans la partie plus orientée « interfaces graphiques »), enchaînant avec la suite des notions de base de la programmation « *mais cette fois entièrement orientée objets* » et s'accompagnent d'exercices permettant d'illustrer cette théorie de base de l'orienté objets (et de la conception d'interfaces graphiques). Chaque cours est accompagné d'un support de cours du sujet du jour, d'exemples complets illustrant les techniques d'implémentation et d'exercices en classe. La matière est systématiquement « démystifiée » en classe par le chargé de cours lors de chaque séance « à thème ». Chaque thème du jour exposé étant un prérequis au thème suivant.

Suite à chaque exposé en classe et progressivement également, il est demandé aux étudiants de constituer un référentiel individuel de solutions aux exercices « à rapport direct avec chaque cours » et qui sont « proposés » par le chargé de cours. Il est « vivement » conseillé aux étudiants que ces solutions soient rendues au chargé de cours « à la date convenue ». Ces solutions peuvent ainsi être commentées « individuellement » (voire évaluées) par le chargé de cours (si rendues à la date convenue) et l'étudiant peut, de son côté, continuer d'avancer ainsi progressivement dans la matière en visant à s'assurer de fixer les « savoirs » via un certain « savoir-faire » en programmation « orientée objets » ainsi que dans la l'implémentation d'IHM !

Sur base du bagage des connaissances durant ce cours, des exercices plus élaborés seront proposés aux étudiants. Ces exercices intégreront le « savoir-faire » technique à des logiques métiers. Cette catégorie d'exercices seront du niveau attendu en terme d'évaluation au final et seront « à préparer et à rendre » lors de l'évaluation finale.

## MODES D'ÉVALUATION

Capacités terminales : <http://admin.segec.be/documents/7041.pdf> (pg.50,51 : capacités terminales)

Pour atteindre le seuil de réussite l' étudiant sera capable d'effectuer l'analyse informatique, de programmer et de tester une/des applications techniques nécessitant l'emploi :

- ❖ des concepts de base de la programmation orientée objet (encapsulation, héritage, polymorphisme,...),
- ❖ d'une bibliothèque de classes pour l'accès à des bases de données relationnelles locales,

- ❖ d'une bibliothèque de classes pour la création de graphiques et pour l'interfaçage graphique des programmes.

Pour la détermination du degré de maîtrise, il sera tenu compte des critères suivants :

- ❖ la pertinence de l'analyse,
- ❖ la fiabilité des tests,
- ❖ l'optimisation des structures de données, des fichiers et des programmes,
- ❖ la présentation et la documentation des programmes,
- ❖ le degré d'autonomie atteint.

Pour l'évaluation :

- une première partie de l'évaluation se fera sur base d'exercices spécifiques qui seront à préparer et à rendre lors de l'évaluation finale ;
- une seconde partie de l'évaluation consistera en une série d'exercices portant sur la connaissance de la matière « savoir théorie » et la capacité à implémenter « savoir-faire », le tout à réaliser dans un temps déterminé le jour de l'évaluation finale.

## **ANNEXE(S)**

→ Dossier pédagogique de l'UE de la section

- <http://admin.segec.be/documents/7041.pdf> (pages 52 à 55)