

IDENTIFICATION	
Unité de formation : PROGRAMMATION ORIENTEE OBJET Activité d'enseignement : PROGRAMMATION ORIENTEE OBJET Cours : PROGRAMMATION ORIENTEE OBJET	Section :INFA Nombre de crédits ECTS : 9
Nombre de périodes : 120	N°UF : 1100
Dossier pédagogique : 7525 21 U32 D2	Unité déterminante : NON

DESCRIPTION
<p><u>Capacités préalables requises :</u></p> <p>Capacités</p> <p><i>En disposant d'une structure informatique matérielle et logicielle opérationnelle et d'une documentation appropriée, face à un problème mettant en jeu des algorithmes de base, dans le respect du temps imparti,</i></p> <ul style="list-style-type: none"> ◆ mettre en oeuvre une stratégie cohérente de résolution du problème posé ; ◆ concevoir, de construire et de représenter l' (les) algorithm(e)s correspondant(s) ; ◆ développer des programmes en respectant les spécificités du langage choisi ; ◆ justifier la démarche algorithmique et les choix mis en oeuvre. <p>Titre pouvant en tenir lieu</p> <p>Attestation de réussite de l'unité de formation «Principes algorithmiques et programmation », n° de code 7521 05 U32 D2, classée dans l'enseignement supérieur économique de type court.</p>
<p><u>Contenu du cours :</u></p> <p><i>En disposant d'une structure informatique matérielle et logicielle opérationnelle, d'une documentation appropriée, les consignes de réalisation de l'application lui étant précisées, et dans le respect des normes de sécurité, d'hygiène et d'environnement, l'étudiant sera capable de :</i></p> <ul style="list-style-type: none"> ◆ concevoir, identifier et modéliser des objets ; ◆ fournir une représentation de ces objets et des relations entre eux ; ◆ développer une application qui résout un problème posé en mobilisant des connaissances et savoir-faire spécifiques : <ul style="list-style-type: none"> • les structures de données dynamiques, • les classes et les méthodes, • les notions d'héritage, • le polymorphisme, • la surcharge, • l'encapsulation, • les fonctions amies, • la programmation événementielle ; ◆ développer une application : en appréhender la structuration globale et en réaliser les composants ;

◆ mettre en oeuvre un style de programmation et des comportements de type professionnel, notamment :

- la documentation du logiciel,
- le respect des standards de programmation de l'équipe,
- l'emploi de dictionnaires de données,
- l'emploi de bibliothèques de fonctions et de procédures ;
- concevoir et mettre en oeuvre une procédure de test partiel et intégré ;
- recourir à bon escient à la documentation disponible.

Capacités terminales :

Pour atteindre le seuil de réussite, l'étudiant devra prouver qu'il est capable :

En disposant d'une structure informatique matérielle et logicielle opérationnelle, d'une documentation appropriée, les consignes de réalisation de l'application lui étant précisées,

- ◆ de concevoir, d'installer et d'utiliser des objets appropriés à la solution ;
- ◆ de concevoir et mettre en oeuvre une procédure de test partiel et intégré ;
- ◆ de justifier sa méthode de résolution ainsi que ses choix conceptuels et méthodologiques.

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

- ◆ de la rigueur et du respect des spécificités du langage de programmation utilisé ;
- ◆ du style de programmation ;
- ◆ de l'adéquation et de la pertinence de la solution développée ;
- ◆ du respect du temps alloué ;
- ◆ de la clarté et de la précision dans l'utilisation du vocabulaire technique ;
- ◆ du degré d'autonomie atteint.

PERSONNEL ENSEIGNANT

MARTENS Fernand

METHODOLOGIE

Sur la plate-forme Claroline, les étudiants peuvent disposer de notes de cours au format PDF ainsi que des exemples de programmes.

Langage utilisé : Java

MODES D'ÉVALUATION

L'évaluation continue est réalisée sous forme d'interrogations et/ou de devoirs à difficultés croissantes. Un examen en fin de module est organisé sur ordinateur.