

| | | | |
|---------------------------|--|---------------------------|------------------------------|
| Section Bac | Informatique de gestion | | |
| Nom de l'UE | Principes algorithmiques et programmation | | |
| Nom du professeur | B. Verlaine, F. Koeune, C. Masson | | |
| Niveau | 1 | Unité déterminante | Non |
| Nombre de périodes | 120 | Pré requis pour | Programmation orientée objet |
| Langue de cours | français | Crédits ECTS | 8 |

I. PRINCIPES D'ALGORITHMIQUES

Objectifs du cours

- Se familiariser avec le développement informatique (*Utiliser les termes adéquats ; Différencier les concepts d'algorithme, de programme et de spécifications ; Avoir une vision simple du processus de développement logiciel*)
- Savoir réaliser les spécifications d'un problème simple (*Comprendre un problème de développement applicatif ; Analyser le problème pour différencier les interactions et l'environnement ; Spécifier la solution au problème*)
- Savoir utiliser les structures algorithmiques standards (*Utiliser des structures alternatives et itératives ; Créer et utiliser des sous-programmes ; Créer et lire des fichiers*)

Contenu du cours

- Partie I: Concepts de base
 1. Introduction
 2. Programmer dans un langage particulier
 3. Réaliser les spécifications d'un programme
 4. Spécifier les interactions
- Partie II: Les bases de l'algorithmique
 1. Base et structure
 2. Algorithmique: Instructions élémentaires
 3. La composition alternative
 4. La composition itérative
 5. Imbrication d'instructions
- Partie III: Algorithmique procédural avancé
 1. Réaliser l'invariant d'une boucle
 2. La notion de tableau
 3. La notion de pile
 4. La notion de sous-programme: la procédure et la fonction
 5. Enregistrer et lire des données depuis un fichier

Méthodologie

Toutes les séances de cours se déroulent en auditoire. Chaque séance comprend des explications théoriques illustrées par des exemples concrets et des extraits de code source en C#, et des exercices réalisés sur papier qui sont corrigés collégalement sur base d'une proposition d'un étudiant

Supports

Les étudiants disposent d'un syllabus composé principalement des slides projetés durant les séances. Ils disposent également de certains corrigés mis en ligne sur la plate-forme électronique destinée aux étudiants.

Acquis d'apprentissage

Pour atteindre le seuil de réussite, l'étudiant sera capable :

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,

- De mettre en œuvre une stratégie cohérente de résolution du problème proposé ;
- De concevoir, de construire et de représenter l' (les) algorithme(s) correspondant(s) ;
- De justifier la démarche algorithmique et les choix mis en œuvre ;
- De développer des programmes en respectant les spécificités du langage choisi ;
- De mettre en œuvre des procédures de test

Evaluation

Examen écrit à livre fermé. 25% des points portent sur la théorie et 75% sur des exercices (lecture et réalisation de spécifications, lecture et réalisation d'algorithmes).

Pondération

| | |
|---|---|
| Evaluation de 1ere session | Evaluation de 2de session |
| Examen écrit à livres fermés (25% pour la théorie, 75% pour les exercices) | Examen écrit à livres fermés (25% pour la théorie, 75% pour les exercices) |

II C# (Fr. Koeune)

Objectifs du cours

- Aborder les éléments de base du langage C#
- Aborder l'environnement de développement DevStudio
- Apprendre à écrire des programmes simples (test de primalité d'un entier, saisie et traitement de données, ...)

Contenu du cours

- Environnement .Net et C#
- syntaxe, types de base,
- exécution conditionnelle, boucles, ...
- méthodes
- Environnement DevStudio : édition, compilation, exécution, debugging

Méthodologie

- Cours ex-cathedra réduit (25% du temps)
- Exercices dirigés en salle machine

Supports

- Livre de référence : C# et .Net versions 1 à 4, de G. Leblanc
- Aide en ligne
- Slides du cours

Acquis d'apprentissage

Pour atteindre le seuil de réussite, l'étudiant sera capable :

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,

- De développer des programmes en respectant les spécificités du langage choisi ;
- De mettre en œuvre des procédures de test

Evaluation

Examen sur machine

Pondération

| | |
|----------------------------|---------------------------|
| Evaluation de 1ere session | Evaluation de 2de session |
| Examen sur machine | Examen sur machine |

III. JAVA (M. Bernair et Cl.Masson)

Objectifs du cours

- Se familiariser avec l'environnement de développement Java (JRE, JDK, Eclipse)
- Connaître les éléments de base de Java : les variables (de type simple ou « primitif » et de type complexe ou des « objets »), les opérateurs (arithmétiques, de conversion, ...), les entrées et sorties standards, les conditions, les boucles, les tableaux, les méthodes.
- Savoir écrire des programmes en Java.

Contenu du cours

- Introduction :
 - caractéristiques de Java
 - bref historique
 - différentes éditions et versions de Java
 - différences entre Java et JavaScript
 - installation du JDK et Eclipse
 - définition d'un langage de programmation, et introduction aux principes d'un langage de programmation orientée objet.
- Règles d'écriture du code Java
- Les variables et les opérateurs
- Lecture des entrées clavier

- Les conditions
- Les boucles
- Les tableaux
- Les méthodes

Méthodologie

- Cours ex-cathedra réduit (25% du temps)
- Exercices dirigés en salle machine

Supports

- Livre de référence : Programmer en Java, de Claude Delannoy, 8ième édition
- Aide en ligne
- Slides du cours au format pdf

Acquis d'apprentissage

Pour atteindre le seuil de réussite, l'étudiant sera capable :

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,

- De développer des programmes en respectant les spécificités du langage choisi ;
- De mettre en œuvre des procédures de test

Evaluation

Examen sur machine

Pondération

| | |
|----------------------------|---------------------------|
| Evaluation de 1ere session | Evaluation de 2de session |
| Examen sur machine | Examen sur machine |