

Programme de Master 1 Informatique : Semestre 7

Codes couleurs :

EN ROUGE = Cours auxquels je présente vraiment des difficultés (compréhension, mise en pratique), cours à travailler en priorité

EN ORANGE = Cours auxquels je présente quelques difficultés, à essayer de travailler si possible

EN VERT = Cours ayant déjà été abordés, mais auquel je ne présente pas de difficultés majeures, à travailler que dans un second temps

Les cours sans couleur sont ceux qui n'ont pas encore abordé, donc inutile de travailler dessus pour le moment.

Cette liste sera régulièrement mise à jour dès que les nouveaux cours seront abordés.

► Complexité (coefficient 6/30) :

- Rappels sur les notions d'analyse de la complexité des algorithmes et des problèmes.
- Notion de problèmes (décision, recherche, dénombrement, énumération, optimisation). Bornes inférieures de complexité.
- Classes P et NP. Problèmes NP-complets.
- Méthodes de résolution des problèmes NP-Complets.

► Algorithmique et recherche opérationnelle (coeff 3/30) :

- Programmation linéaire.
- Notions et algorithmes de flots.
- Applications (couplage, affectation, transport, ...)
- Programmation dynamique (sac à dos, ordonnancement, ...)
- Algorithmes à performance garantie

► Génie logiciel (coeff 6/30) :

1) Versioning

- Git: Branches, Dépôts Lointains, Dépôts Lointains multiples avec branches, Workflows.

2) Building

- Gradle: Automatisation de compilation, de tests, de dépendances, langage de tâches interdépendantes.

3) Méthodes

- Les acteurs du logiciel.
- Cycles en V. Motivation et détail de chacune des phases. Forces et Faiblesses.
- Méthodes agiles. Motivations. Notamment XP et Scrum. Détail de chacune des réunions et instruments. Forces et Faiblesses.
- GL Libre: the cathedral and the bazaar

4) Documentation

- Du GL: Cahier des charges, Spécifications, Conception, ...
- Du code: Javadoc, Commentaires, Wikis, Issues trackers

5) UML

- Tous les principaux diagrammes UML (classe, object, séquence, com, package, états-transitions,...).

6) Tests

- Types de Test (Blanche/Noire, Intégration (continue), Correction/Validation,...), Couverture (Exos sur les différents critères), Quand les faire (TDD...)
- Découverte du debugger eclipse
- Concepts d'invariants, d'assertions, d'analyse statique

7) Estimation

- Rapports PrixDev vs TempsDev vs LignesCode
- % observés de failed projects, causes observées
- % observés de bugs

8) Veille technologique sur l'actualité du GL

► Réseaux (coeff 4/30) :

- Architecture en couches, standard OSI
- Couche physique
- Couche liaison de Données
- Réseau IP
- Algorithmes de routage, routage IP
- Protocole TCP/IP
- Couche Application : HTTP, SMTP, FTP, ...
- Bases des protocoles cryptographiques (terminologie, architecture à clef publique).
- Introduction à la Sécurité des réseaux.

► Programmation objet concurrente (coeff 3/30):

- Threads en Java
- Problème d'atomicité et techniques classiques de synchronisation
- Collections concurrentes, réservoirs de threads, etc
- Programmation optimiste et programmation sans verrou
- Le modèle mémoire Java

► Science des données (coeff 3/30):

- Algèbre linéaire
- Clustering : apprentissage non supervisé, ACP, kmeans, visualisation
- Classification : apprentissage supervisé, notion d'erreur et modèle, erreur de Bayes, kppv, généralisation et sur-apprentissage.
- Régression : notions de causalité, notion d'erreur, protocoles d'évaluation.
- Sensibilisation aux aspects éthiques (GAFAM).
- Utilisation d'approches probabilistes (Bayes), statistiques, modes d'évaluation et visualisation.