

## Informatique 2 pour microtechniciens

<b>Domaine</b>	Ingénierie et Architecture
<b>Filière</b>	Microtechniques
<b>Orientation</b>	Microtechniques (MI)
<b>Mode de formation</b>	Temps partiel/En emploi

### Informations générales

Nom:	:	Informatique 2 pour microtechniciens
Identifiant:	:	Info2Mi
Années académiques	:	2018-2019, 2019-2020, 2020-2021
Responsable:	:	Pierre Bressy
Charge de travail:	:	120 heures d'études
Périodes encadrées:	:	80 (= 60 heures)

Semestre	E1	S1	S2	E2	S3	S4	E3	S5	S6	E4	S7	S8
Cours						48						
Laboratoire						32						

### Connaissances préalables recommandées

L'étudiant-e doit connaître et savoir utiliser les notions suivantes :

- Bases de la programmation : types de base, structures de contrôle et sous-programmes.

L'unité d'enseignement Info1MI (Informatique 1 pour Microtechniciens) permet d'acquérir ces connaissances.

### Objectifs

A l'issue de cette unité d'enseignement, l'étudiant-e sera capable de :

- Décomposer un algorithme selon l'approche descendante (raffinage successif) et ascendante.
- Décomposer une application de complexité moyenne en algorithmes élémentaires.
- Concevoir un type de données abstrait simple et les fonctions pour le manipuler.
- Concevoir une bibliothèque de fonctions en utilisant la compilation séparée.
- Ecrire un programme qui lit ou écrit un fichier au format binaire, texte.
- Mettre en œuvre l'allocation dynamique de mémoire pour créer des tableaux dimensionnés à l'exécution.
- Définir et manipuler un type de données récursif (liste chaînée).
- Mettre en œuvre des champs de bits et des unions pour manipuler des bits dans un masque binaire.

A l'issue des travaux pratiques en laboratoire, principalement destinés à l'assimilation des connaissances et à l'acquisition d'expérience professionnelle, l'étudiant-e sera en outre capable de :

- Programmer une bibliothèque de fonctions de qualité professionnelle (et l'utiliser).
- Programmer et mettre au point des algorithmes de complexité moyenne.
- Réaliser une application de taille et de complexité moyennes, mêlant différents aspects de la programmation.

## Contenu et formes d'enseignement

*Répartition des périodes indiquée à titre informatif.*

**Cours:** 48 périodes

- Préprocesseur. 4
- Classes de stockage. 4
- Conception de types de données abstraits simples, création de bibliothèques. 8
- Pointeurs, allocation dynamique de mémoire. 10
- Implémentation des listes, files et piles basée sur les tableaux 6
- Types de données récurifs :files, piles 8
- Traitement des fichiers binaires, textes. 4
- Contrôle continu. 4

**Laboratoire:** 32 périodes

- Mise en oeuvre de types de données composés (structures, tableaux multidimensionnels) 6
- Lecture et écriture de fichiers texte et binaire en mode séquentiel 4
- Mise en oeuvre de l'allocation dynamique de mémoire 4
- Compilation séparée et implémentation de bibliothèques. 2
- Implémentation de types de données abstraits : type simple, liste tableau. 4
- Implémentation de types de données récurifs : file, pile. 6
- Mini-projet. 6

## Bibliographie

- Jean-Michel Léry, Algorithmique en C, 2e édition, Pearson, 2010
- Jean-Michel Léry, Algorithmique, Applications en C, C++ et Java, Pearson, 2013
- Thomas Cormen, Introduction à l'algorithmique, 3ème édition, Dunod, 2010

## Contrôle de connaissances

**Cours:**

l'acquisition des matières de cet enseignement sera contrôlée au fur et à mesure par des tests et des travaux personnels tout au long de son déroulement. Il y aura au moins 2 tests d'une durée totale d'au moins 3 périodes.

**Laboratoire:**

ils seront évalués sur la base des rapports de manipulation, à 3 reprises au minimum.

**Examen:**

L'atteinte de l'ensemble des objectifs de formation sera vérifiée lors d'un contrôle final commun écrit d'une durée de 90 minutes.

**Matériel autorisé:**

- Information communiquée directement par l'enseignant.

**Calcul de la note finale**

Note finale = moyenne cours x 0.25 + moyenne laboratoire x 0.25 + moyenne examen x 0.5

Fiche validée le 12.09.2018 par Schmitt Carl