



# Présentation

Anicet E. T. Ebou, [ediman.ebou@inphb.ci](mailto:ediman.ebou@inphb.ci)



Ce travail est soumis à une licence internationale Creative Commons Attribution 4.0.



# Salut!

**Anicet E. T. Ebou**

Enseignant-Chercheur à l'INP-HB.

Anicet est un bioinformaticien qui étudie les microbiomes et la génomique computationnelle avec un amour particulier pour la conception d'algorithmes et d'outils pour l'analyse biologique.

**Email:** [ediman.ebou@inphb.ci](mailto:ediman.ebou@inphb.ci) | **Twitter / X:** [@anicetebou](https://twitter.com/anicetebou)

**Bureau:** 130 - DFR ARA - INP-HB site Nord

# Qu'est ce qu'un ingénieur?

Les ingénieurs, en tant que praticiens de l'ingénierie, sont des professionnels qui inventent, conçoivent, analysent, construisent et testent des machines, des systèmes complexes, des structures, des gadgets et des matériaux pour remplir des objectifs et des exigences fonctionnels tout en tenant compte des limites imposées par l'aspect pratique, la réglementation, la sécurité et le coût.

<https://en.wikipedia.org/wiki/Engineer>

# Objectif du programme

La formation de futurs **ingénieures et ingénieurs**, vétérinaires, enseignantes et enseignants, chercheuses et chercheurs et avant tout des **personnes informées, capables de gouverner** leur vie professionnelle et citoyenne en **pleine connaissance et maîtrise des techniques et des enjeux de l'informatique** et en la nourrissant par les **habitudes de la démarche scientifique**.

# Compétences visées

1

**Analyser et modéliser** un problème ou une situation, notamment en utilisant les objets conceptuels de l'informatique pertinents (table relationnelle, graphe, dictionnaire, etc.).

# Compétences visées

## 2

**Imaginer et concevoir une solution**, décomposer en blocs, se ramener à des sous-problèmes simples et indépendants, adopter une stratégie appropriée, décrire une démarche, un algorithme ou une structure de données permettant de résoudre le problème.

# Compétences visées

## 3

**Décrire et spécifier** les caractéristiques d'un processus, les données d'un problème, ou celles manipulées par un algorithme ou une fonction.

## 4

**Mettre en œuvre une solution**, par la traduction d'un algorithme ou d'une structure de données dans un langage de programmation ou un langage de requête.

## Compétences visées

5

**Justifier et critiquer une solution**, en développant des processus d'évaluation, de contrôle, de validation d'un code que l'on a produit.

6

**Communiquer à l'écrit ou à l'oral**, présenter des travaux informatiques, une problématique et sa solution ; défendre ses choix ; documenter sa production et son implémentation.



# Apprentissage et évaluation

- Cours magistral + Travaux Dirigés + Travaux Pratiques
- Exercices de travaux pratiques évalués (c.à.d. notés)
- Evaluation continue (c.à.d. Interrogations, voir la progression)
- Test léger à la fin de chaque module
- Devoir de niveau programmé par l'école
- Votre évaluation dans ce module est constituée de:
  - 30% TP et interrogations
  - 30% de tests légers
  - 40% de devoir de niveau

# Programme semestre 1

## **Module 1: Architecture des ordinateurs**

- **Chapitre 1:** Historique de l'ordinateur et ses composantes
- **Chapitre 2:** Représentation des données
- Travaux dirigés 1
- Test léger 1

# Programme semestre 1

## Module 2: Algorithmique et programmation Python

- **Chapitre 1:** Introduction aux algorithmes
- **Chapitre 2:** Variables en algorithmique
- Travaux dirigés 2
- **Chapitre 3:** Algorithmes élémentaires opérants par structure alternative
- **Chapitre 4:** Algorithmes élémentaires opérants par boucles simples
- Travaux dirigés 3
- Interrogation 3

# Programme semestre 1

## Module 2: Algorithmique et programmation Python

- **Chapitre 5:** Introduction à Python
- **Chapitre 6:** Algorithmes opérants sur les tableaux unidimensionnels
- **Chapitre 7:** Algorithmes opérants par boucles imbriquées
- **Chapitre 8:** Présentation de la bibliothèque NumPy
- Travaux dirigés et pratiques 4
- Interrogation 4

# Programme semestre 1

## Module 2: Algorithmique et programmation Python

- **Chapitre 9:** Procédures, fonctions et fonctions récursives
- **Chapitre 10:** Algorithmes de tris naïfs
- Travaux dirigés 5
- **Travaux pratiques:** Matrices de pixels et d'images
- **Travaux pratiques:** Gestion des fichiers
- Test léger 2

# Programme semestre 2

## Module 3: Méthodes de programmation et dictionnaire

- **Chapitre 1:** Notion de mutabilité et effets de bords
- **Chapitre 2:** Tableau associatif ou dictionnaire
- Travaux dirigés 6

## Module 4: Bases de données

- **Chapitre 1:** Vocabulaires, requêtes et applications
- Travaux dirigés 7
- Interrogation 5
- Test léger 3

# Programme semestre 2

## Module 5: Graphes

- **Chapitre 1:** Notions de base et modélisation de graphe
- **Chapitre 2:** Algorithme de parcours en largeur
- Travaux dirigés 8
- Interrogation 6

# Programme semestre 2

## Module 6: Méthodes numériques

- **Travaux Pratiques:** Calcul d'intégrales avec Python
- **Chapitre 1:** Modélisation d'expériences aléatoire et estimation de distribution de loi
- **Travaux Pratiques:** Simulation d'expérience aléatoire
- Travaux dirigés
- Test léger 4



# Comment puis-je réussir mes classes prépa?

1. Exercez-vous dans toutes les matières
2. Ne soyez pas déçu par vos notes (vous aurez beaucoup de zéros)
3. Soyez déterminé
4. Gérez bien votre stress
5. Pratiquez une activité sportive
6. Mettez en place une bonne organisation d'études
7. Optez pour le travail en groupe (pas forcément en "famille")

# Mon avis sur la gestion de votre temps

- **8 heures** de sommeil non négociables par jour
- **8 heures** de cours imposés par l'école par jour
- $8 + 8 - 24 = 8$  **heures que vous devez gérer avec intelligence**
- 30 min de sport, 1 h 30 pour tout ce qui est personnel (toilette, chanter, prier, méditation, rire, etc.), 30 minutes pour manger.
- **5 h 30 de bosse/recherche/travail par jour = 38h 30 minutes par semaine.**

# Les règles de la classe

1. Respect
2. Assiduité
3. Responsabilité