



## Master 1 IA

### Examen Pratique de Python

Durée : 3 semaines — du 19 Decembre 2025 au 09 Janvier 2026

## Contexte général

Le **Dakar Institute of Technology (DIT)** dispose d'une bibliothèque académique destinée aux étudiants, enseignants et au personnel administratif.

Actuellement, la gestion de cette bibliothèque est effectuée de manière **entièrement manuelle** : enregistrement des livres sur cahier, suivi approximatif des emprunts, absence d'historique fiable et difficultés à produire des statistiques.

Face à ces limites, la direction du DIT a décidé de **digitaliser l'ensemble du processus de gestion de la bibliothèque**. Vous êtes recruté(e) en tant que **développeur Python** pour concevoir une application fiable, modulaire et évolutive permettant d'automatiser toutes les opérations.

L'application devra être développée en **Python**, en respectant rigoureusement les principes de la **programmation orientée objet (POO)** et en assurant la **persistance des données** à l'aide de fichiers.

## Objectifs pédagogiques

Cet examen vise à évaluer votre capacité à :

- Modéliser un problème réel à l'aide de classes et d'objets
- Mettre en œuvre l'encapsulation, l'héritage et le polymorphisme
- Structurer une application Python de manière claire et maintenable
- Gérer la persistance des données avec des fichiers
- Appliquer des règles métiers réalistes

## Fonctionnalités à implémenter

### 1 Gestion des livres

- Ajouter, modifier et supprimer un livre

- Lister l'ensemble des livres
- Gestion des exemplaires multiples
- Gestion du numéro ISBN (unique)
- Statut du livre : disponible, emprunté, réservé, perdu, endommagé
- Compteur du nombre d'emprunts par livre

## 2 Gestion des utilisateurs

- Ajouter et lister les utilisateurs
- Types d'utilisateurs :
  - Étudiant
  - Enseignant
  - Personnel administratif
- Limite du nombre d'emprunts selon le type d'utilisateur
- Génération automatique d'un identifiant unique
- Historique complet des emprunts par utilisateur

## 3 Gestion des emprunts

- Emprunter et retourner un livre
- Vérification automatique de la disponibilité d'un livre
- Gestion de la date d'emprunt et de retour prévue
- Détection des retards
- Renouvellement d'emprunt
- Gestion des pénalités en cas de retard

## 4 Système de réservation

- Réserver un livre indisponible
- Gestion d'une file d'attente des réservations
- Notification via fichier texte lorsqu'un livre devient disponible

## 5 Recherche avancée

- Recherche par titre, auteur, catégorie
- Recherche par ISBN et année de publication
- Recherche par disponibilité
- Recherche par mots-clés

## 6 Sauvegarde et journalisation

- Sauvegarde des données dans des fichiers :
  - livres
  - utilisateurs
  - emprunts
  - reservations
- Sauvegarde automatique après chaque opération
- Fichier log enregistrant toutes les actions

## 7 Rapports et statistiques

- Nombre total de livres et de livres disponibles
- Nombre de livres empruntés, réservés, perdus ou endommagés
- Top 5 des livres les plus empruntés
- Top 5 des utilisateurs les plus actifs
- Nombre total d'emprunts effectués
- Liste des livres jamais empruntés

## Livrables attendus

Les éléments suivants sont **obligatoires** et doivent être envoyés à l'adresse email **diopmadicke351@gmail.com** avant la date limite :

- **Rapport détaillé du projet** (format PDF) comprenant :
  - Présentation générale de la solution proposée
  - Analyse du problème et modélisation (diagrammes simples acceptés)
  - Description des classes, attributs et méthodes principales
  - Explication détaillée des étapes de développement
  - Captures d'écran illustrant l'exécution de l'application et les principales fonctionnalités
- **Code source complet de l'application** :
  - Organisation claire des fichiers et dossiers
  - Code lisible, commenté et conforme aux principes de la POO
- **Fichier README** (obligatoire) expliquant clairement :
  - Les prérequis nécessaires (version de Python, bibliothèques utilisées)
  - La structure du projet
  - Les étapes pour installer, configurer et exécuter l'application
  - Les principales fonctionnalités disponibles

## Barème de notation (20 points)

- **Analyse du problème et modélisation POO** : 3 points
  - Pertinence des classes et relations
  - Utilisation correcte de l'encapsulation et de l'héritage
- **Implémentation des fonctionnalités** : 8 points
  - Gestion des livres : 2 points
  - Gestion des utilisateurs : 2 points
  - Gestion des emprunts et réservations : 3 points
  - Recherche et statistiques : 1 point
- **Persistance des données et journalisation** : 2 points
  - Sauvegarde des fichiers fonctionnelle
  - Fichier de log exploitable
- **Qualité du code et organisation du projet** : 3 points
  - Lisibilité, commentaires, structuration
  - Respect des bonnes pratiques Python
- **Rapport et documentation (README)** : 4 points
  - Clarté des explications
  - Qualité des captures d'écran et justification des choix

**Bonne chance à toutes et à tous !**

*Que ce projet soit l'occasion de démontrer votre  
créativité, votre rigueur et votre maîtrise de  
la programmation orientée objet en Python.*

**Travaillez avec sérieux, méthode et passion.**