

CAHIER DES CHARGES

TRAVAUX PRATIQUES - PYTHON

Système de Gestion des Notes Scolaires

Institut Universitaire Catholique Sainte Thérèse de Yaoundé
École Polytechnique Supérieure / VOGT HIGH TECH

📄 PRÉSENTATION DU PROJET

Vous devez développer une **application console en Python** pour gérer les notes des étudiants d'un établissement scolaire. L'application doit permettre la gestion complète des étudiants, des matières et des notes, ainsi que le calcul de statistiques académiques.

Contrainte principale : Le stockage des données doit se faire **exclusivement via des fichiers** (pas de base de données). Vous devez faire preuve d'ingéniosité dans votre approche.

📄 EXIGENCES FONCTIONNELLES OBLIGATOIRES

1. Authentification (2 points)

- Le programme doit démarrer par une demande de mot de passe
- L'accès au système n'est autorisé qu'après validation du mot de passe
- Limiter les tentatives de connexion
- Gérer les erreurs de saisie

2. Gestion des Étudiants (4 points)

Le système doit permettre de :

- Ajouter un étudiant (nom, prénom, matricule unique, niveau/classe)
- Modifier les informations d'un étudiant
- Supprimer un étudiant
- Rechercher un étudiant
- Lister les étudiants (par classe ou tous)

3. Gestion des Matières (3 points)

Le système doit permettre de :

- Ajouter une matière (nom, code, coefficient, niveau/classe concerné)
- Modifier une matière
- Supprimer une matière
- Lister les matières par niveau/classe
- Associer les matières aux classes appropriées

4. Saisie des Notes (5 points)

Le système doit permettre de :

- Enregistrer les notes des étudiants par matière et par niveau
- Valider que les notes sont comprises entre 0 et 20
- Modifier les notes existantes
- Supprimer des notes
- Gérer les cas particuliers (étudiant sans note, etc.)

Note : Un étudiant ne peut avoir qu'une seule note par matière.

5. Consultation des Notes (3 points)

Le système doit permettre de consulter :

- Les notes d'un étudiant spécifique
- Les notes d'une classe pour une matière donnée
- Le bulletin complet d'un étudiant (avec toutes ses notes et moyennes)

L'affichage doit être clair et bien formaté.

6. Statistiques et Analyses (5 points)

Le système doit calculer et afficher :

Par étudiant :

- Moyenne générale (notes pondérées par les coefficients)
- Rang dans sa classe

Par classe :

- Moyenne générale de la classe
- Meilleur étudiant de la classe
- Classement des étudiants

Par matière :

- Moyenne générale par matière
- Meilleur étudiant par matière

Globalement (toute l'école) :

- Moyenne générale de l'établissement
- Meilleur étudiant de l'école

☒ LIBERTÉ CRÉATIVE

Organisation du Menu

Vous êtes **totale**ment **libre** d'organiser l'interface et les menus de votre application. Soyez créatif et proposez une navigation intuitive.

Gestion des Fichiers

Le stockage via fichiers est **obligatoire**. Le choix du format et de l'organisation est **libre**. Faites preuve d'ingéniosité pour :

- Organiser vos données de manière efficace
- Éviter les pertes de données
- Faciliter la lecture et l'écriture
- Gérer la cohérence des informations

☒ BARÈME D'ÉVALUATION

Total : /30 points

Critère	Points
Fonctionnalités Obligatoires	22 pts
1. Authentification	2 pts
2. Gestion des étudiants	4 pts
3. Gestion des matières	3 pts
4. Saisie des notes	5 pts
5. Consultation des notes	3 pts
6. Statistiques complètes	5 pts
Qualité du Code	5 pts
- Structure et organisation	2 pts
- Utilisation de la POO	1 pt
- Gestion des exceptions	1 pt
- Lisibilité et commentaires	1 pt
Gestion des Fichiers	3 pts
- Ingéniosité du système	1 pt

- Fiabilité (pas de corruption) Critère	Points
- Persistance des données	1 pt

📁 LIVRABLES

1. Code Source

- Tous les fichiers `.py` nécessaires au fonctionnement
- Code correctement indenté et commenté

2. Fichiers de Données Exemples

Fournir des données de test pré-remplies :

- Au moins 10 étudiants (répartis sur 2-3 classes minimum)
- Au moins 5 matières
- Au moins 30 notes
- Fichier de configuration avec mot de passe

3. Documentation (README.txt ou README.md)

Votre README doit contenir :

- Votre nom, matricule et classe
- Instructions d'installation et d'exécution
- Mot de passe par défaut
- Description de votre organisation des fichiers
- Fonctionnalités implémentées

4. Format de Soumission

- **Archive ZIP** nommée : `NOM_Prenom_Matricule_TP_Python.zip`
- Tous les fichiers nécessaires doivent être inclus
- Le programme doit être **immédiatement exécutable** après extraction
- Proposer une courte vidéo d'illustration du fonctionnement de votre application

📁 MODALITÉS

Date Limite

[À définir par l'enseignant]