

TP : Entrées/Sorties en Java

- ❖ Manipuler les entrées/sorties standards (Scanner, System.in, System. Out)
- ❖ Lire et écrire dans un fichier texte (FileReader, FileWriter, BufferedReader, BufferedWriter)
- ❖ Utiliser la sérialisation pour enregistrer un objet (ObjectOutputStream, ObjectInputStream)

🔍 Partie 1 : Entrées/Sorties Standard

Exercice 1 : Lecture et affichage des données

✍ Écrire un programme qui :

1. Demande à l'utilisateur d'entrer son **nom** et son **âge**.
2. Affiche ces informations.

Entrez votre nom : mohammed

Entrez votre âge : 33

Bonjour Mohammed, vous avez 33 ans.

Partie 2 : Lecture et Écriture dans un Fichier

Exercice 2 : Écriture dans un fichier

✍ Écrire un programme qui :

1. Demande à l'utilisateur d'entrer un **texte**.
2. Sauvegarde ce texte dans un fichier "**output.txt**".

Exercice 3 : Lecture depuis un fichier

✍ Écrire un programme qui :

1. Lit le contenu du fichier "**output.txt**" et l'affiche.

Sérialisation

Objectif de la Sérialisation

La **sérialisation** en Java permet de **convertir un objet en un flux de bytes** pour :

1. **Sauvegarder un objet dans un fichier.**
2. **Envoyer un objet sur un réseau.**
3. **Stocker un état d'un objet et le restaurer plus tard.**

L'inverse de la sérialisation est la **désérialisation**, qui permet de **reconstruire l'objet** à partir du fichier ou du flux.

Exercice 4 : Sauvegarde et chargement d'un objet

🔗 Créer une classe `Personne` qui :

1. Stocke un **nom** et un **âge**.
2. Peut être enregistrée dans un fichier (`ObjectOutputStream`).
3. Peut être chargée depuis un fichier (`ObjectInputStream`).

Exercice 4 : Manipulation des fichiers (Entrées/Sorties)

Consignes :

1. Créez un fichier texte `data.txt` et écrivez-y quelques lignes de texte.
2. Lisez le contenu de ce fichier et affichez-le sur la console.
3. Copiez le contenu de `data.txt` vers un autre fichier `copy.txt`.