



# Laboratoire d'Innovation Libre et Open Source

## SynaptikOne

### À propos de SynaptikOne :

SynaptikOne est un laboratoire d'innovation libre et open source, *totalelement indépendant*. Nous ne sommes rattachés à aucune organisation ou institution. Notre mission est d'accompagner et d'aider tous les étudiants, qu'ils soient inscrits dans une université ou autodidactes. Nous offrons un environnement ouvert et collaboratif, afin que chacun puisse apprendre, expérimenter et innover librement.

**Nous sommes également ouverts à toute collaboration, partenariat ou opportunité de projet. N'hésitez pas à nous contacter pour explorer ensemble de nouvelles idées et initiatives.**

**Rédigé par : Rakotondravelo Tahina Mickaël**

Fondateur du Laboratoire SynaptikOne

**Licence :** Creative Commons BY-NC-SA 4.0 — Usage commercial interdit.

**Site web :** <https://synaptikone.pages.dev/>

**Facebook :** Page Facebook SynaptikOne

# Énoncé du sujet - Programmation Java et C++ (Master 1, Parcours Informatique)

Université CNTEMAD — Session 2024–2025

1. Exercice 1 (10 points) – Java : programme de pointage quotidien
2. Exercice 2 (10 points) – C++ : gestion d'un petit stock de produits

## Corrigé détaillé

### Exercice 1 – Java : Pointage quotidien

**Objectif :** Calculer la durée travaillée, le retard par rapport à 09 :00, et les heures supplémentaires au-delà de 8h.

**Logique algorithmique (5 pts) :**

- Saisir l'heure d'entrée et l'heure de sortie (format HH :mm)
- Calculer la durée travaillée : sortie - entrée
- Calculer le retard éventuel : comparer l'heure d'entrée avec 09 :00
- Calculer les heures supplémentaires : durée travaillée - 8h
- Afficher les résultats

**Exemple de code Java (5 pts) :**

```
import java.time.LocalDateTime;
import java.time.Duration;
import java.util.Scanner;

public class Pointage {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Heure entree (HH:mm) : ");
        LocalDateTime entree = LocalDateTime.parse(sc.nextLine());
        System.out.print("Heure sortie (HH:mm) : ");
        LocalDateTime sortie = LocalDateTime.parse(sc.nextLine());

        Duration duree = Duration.between(entree, sortie);
        long heures = duree.toHours();
        long minutes = duree.toMinutes() % 60;
```

```
Duration retard = Duration.between(LocalTime.of(9,0), entree);
long retardMin = Math.max(0, retard.toMinutes());

long heuresSup = Math.max(0, heures - 8);

System.out.println("Duree travaillée : " + heures + "h " + minutes + "min");
System.out.println("Retard : " + retardMin + " minutes");
System.out.println("Heures supplémentaires : " + heuresSup + " heures");
}
}
```

## Exercice 2 – C++ : Gestion d'un stock de produits

**Objectif :** Ajouter, mettre à jour, rechercher un produit et calculer la valeur totale du stock.

### Logique algorithmique (5 pts) :

- Créer une structure ou classe `Produit` avec nom, prix, quantité
- Ajouter un produit dans un vecteur
- Mettre à jour la quantité d'un produit existant
- Parcourir le vecteur pour calculer la valeur totale
- Rechercher un produit par nom et afficher ses informations

### Exemple de code C++ (5 pts) :

```
#include <iostream>
#include <vector>
#include <string>
using namespace std;

struct Produit {
    string nom;
    double prix;
    int quantite;
};

int main() {
```

```
vector<Produit> stock;
Produit p1 = {"Stylo", 1.5, 10};
Produit p2 = {"Carnet", 2.0, 5};
stock.push_back(p1);
stock.push_back(p2);

// Mise à jour
stock[0].quantite += 5;

// Valeur totale
double valeurTotale = 0;
for(auto &p : stock) valeurTotale += p.prix * p.quantite;
cout << "Valeur totale du stock: " << valeurTotale << endl;

// Recherche
string nomRecherche = "Carnet";
for(auto &p : stock){
    if(p.nom == nomRecherche){
        cout << "Produit: " << p.nom << ", Prix: " << p.prix
        << ", Quantite: " << p.quantite << endl;
    }
}
}
```

## Licence

### Licence Creative Commons BY-NC-SA 4.0

Attribution – Pas d’usage commercial – Partage dans les mêmes conditions.

© SynaptikOne, Laboratoire d’Innovation Libre et Open Source.

## Remarque

*Les corrigés proposés par le Laboratoire SynaptikOne sont élaborés avec soin et rigueur. Cependant, en raison du nombre important de sujets, projets et documents traités quotidiennement, il est possible que certains corrigés comportent des erreurs ou des imprécisions. Nous invitons chaque lecteur à nous signaler toute correction ou suggestion*

---

*d’amélioration, afin de contribuer ensemble à l’enrichissement et à la fiabilité de nos ressources pédagogiques.*