
**Master Langue et Informatique**
**Université Paris-Sorbonne**

# MASTER LANGUE ET INFORMATIQUE

Programmation générique et conception objet

## Atelier n° 3
## Surcharge des opérateurs

Les exercices qui vous sont proposés ici ont pour but d'apprendre à utiliser les notions de surcharge d'opérateur.

## 1. MA PREMIERE SURCHAGE D'OPERATEUR

Le but de cette partie est d'écrire des surcharges d'opérateur pour le composant de Date de l'atelier 1. Copier les fichiers Date.h et Date.cpp dans le répertoire correspondant à l'atelier 3.

**Exercice 1** : Modifier le code pour redéfinir l'opérateur binaire ++ pour l'incrémentation d'une date.

**Exercice 2** : Modifier le code pour redéfinir l'opérateur binaire << pour l'affichage d'une date.

**Exercice 3** : Modifier le code pour redéfinir l'opérateur binaire == pour le test d'égalité de deux dates.

## 2. REDEFINITION DES OPERATEURS DE LA CLASSE CARAFE

Soit la classe Java Carafe.java disponible dans le cours de Programmation Objet (cf Annexe). On souhaite réécrire ce programme en C++ pour utiliser les surcharges d'opérateur.

**Exercice 1** : Ecrire les fichiers Carafe.h et Carafe.cpp et test.cpp (contenant la fonction main). Tester le code.

**Exercice 2** : Choisir un ensemble d'opérateurs sémantiquement cohérent pour représenter les méthodes remplir, vider, transvaser.

**Exercice 3** : Modifier les codes de Carafe.h et Carafe.cpp pour surcharger les opérateurs choisis.

**Exercice 4** : Modifier le code de test.cpp

## ANNEXE CLASSE CARAFE.JAVA

```
public class Carafe {
 int contenu;
 final int capacite;

 Carafe(int contenu,int capacite) {
       this.contenu = contenu;
       this.capacite  = capacite;
 }

 public void remplir() {
       contenu = capacite;
 }

 public void vider() {
```

```
        contenu = 0;
    }

    public int contenu() {
        return contenu;
    }

    public int capacite() {
        return capacite;
    }

    public void transvaser(Carafe c) {
        if (((contenu + c.contenu) <= c.capacite)) {
            c.contenu += contenu;
            contenu = 0;
        }
        else {
            contenu -= (c.capacite - c.contenu);
            c.contenu = c.capacite;
        }


    }

}
```