

<b>Séance N°7</b>
-------------------

**Surcharge opérateurs et fonctions amies**

Le C++ permet de redéfinir les opérateurs courants avec d'autres paramètres et d'autres opérations, ce qui permet de les utiliser avec les données membres d'une classe. Le compilateur reconnaît automatiquement la fonction qui convient selon le type des paramètres utilisés.

**Exemple :**

Reprendre le projet réalisé au précédent TD comprenant les 5 fichiers (principal.cpp, CTemperature.h, CTemperature.cpp, CThermocouple.h, CThermocouple.cpp), et ajouter une opération à la classe CTemperature, permettant de soustraire deux températures, en opérant comme suit :

- déclarer l'opération dans le fichier **CTemperature.h**

<pre>... ajouter les déclarations des opérateurs mathématiques dans la classe  double operator - (CTemperature T); // Temperature - T</pre>
---

- définir l'opération dans le fichier **CTemperature.cpp**

<pre>... ajouter la définition des opérations à la fin  double CTemperature::operator - (CTemperature T) // Temperature - T {return Temperature - T.Temperature; }</pre>
--

- tester dans le fichier **principal.cpp**

<pre>... après avoir déclaré les deux variables T1 et T2 de la classe CTemperature, et les avoir initialisées par une des fonctions de Saisie, tester dans le programme principal  cout &lt;&lt; "T1 - T2 = " &lt;&lt; T1 - T2 &lt;&lt; endl;</pre>
---

Recompiler et tester cette nouvelle version.  
Recommencer avec l'opérateur +

**TRAVAIL PERSONNEL**

**Exercice 1 :** en vous inspirant de l'exemple précédent, ajouter les opérations suivantes à la classe **CTemperature**

Opérateur	Fonction	Sortie
/	Quotient de 2 températures	Double
==	Egalité entre 2 températures	Bool
!=	Différence entre 2 températures	Bool
>	Température supérieure à une autre	Bool
<	Température inférieure à une autre	Bool

Vous pourrez les tester l'une après l'autre dans le fichier **principal.cpp** avec ces lignes :

```
... ajouter ces lignes

cout << "T1 / T2 = " << T1 / T2 << endl;

if (T1 > T2) cout << "T1 > T2" << endl;
if (T1 < T2) cout << "T1 < T2" << endl;
if (T1 == T2) cout << "T1 égal à T2" << endl;
if (T1 != T2) cout << "T1 différent de T2" << endl;
```

**Exercice 2 :** ajouter des fonctions à la classe **CTemperature** pour pouvoir effectuer des opérations entre un réel et une température. Dans ce cas, il faut assurer la **commutativité**, c'est à dire pouvoir faire des opérations dans les 2 sens : r+T et T+r, et utiliser des **fonctions amies**. Voici un exemple,

- déclarer dans le fichier **CTemperature.h**

```
... ajouter
friend CTemperature operator + (CTemperature T, double r); // T+r
friend CTemperature operator + (double r, CTemperature T); // r+T
operator double(); // opérateur de conversion vers un double
```

- définir dans le fichier **CTemperature.cpp**

```
... ajouter ces lignes
CTemperature::operator double()
{return Temperature;
}
CTemperature operator + (CTemperature T, double r) // T+r
{CTemperature V;
 V.Temperature = T.Temperature + r;
 return V;
}
CTemperature operator + (double r, CTemperature T) // r+T
{CTemperature V;
 V.Temperature = T.Temperature + r;
 return V;
}
```

- tester dans le fichier **principal.cpp**

```
... ajouter
cout << "T2 + 10.0 = " << T2 + 10.0 << endl;
cout << "5.0 + T3 = " << 5.0 + T3 << endl;
```

Recompiler et tester cette nouvelle version.

Puis, en vous inspirant de l'exemple précédent, ajouter les opérations suivantes de telle sorte qu'elles soient commutatives :

Opérateur	Fonction
-	Soustraction d'une température avec un réel
*	Multiplication d'une température avec un réel

**Règle sur les fonctions amies:** on doit utiliser une fonction amie si l'opération met en jeu 2 opérandes et que le premier opérande n'est pas du type de la donnée membre retournée. Ainsi le préfixe est absolument nécessaire pour effectuer les opérations (r+T) et (r\*T).

#### **BILAN PERSONNEL**

**Ce que j'ai appris aujourd'hui :** (à compléter)

**Vocabulaire informatique :** (à compléter)

