Transtypage

En informatique, le « transtypage » est une opération de **conversion de type** ou ***cast***.

Une conversion de type (ou de promotion de type) peut être **implicite** (automatique) ou **explicite** (c'est-à-dire forcée par le programmeur).

Lorsqu'elle est explicite, on utilise un opérateur de ***cast***. Les conversions forcées peuvent être des conversions dégradantes (avec perte). Cela peut être dangereux (source d'erreur).

Conversion de type forçée (*cast*) :

.c

```
char c = 5;
int a = (int)c;
```

conversion en int

.cpp

```
char c = 5;
int a = int(c);
```

conversion en int

Conversion implicite (avec perte) :

```
int x = 5;
float y = 1.5;
int resultat;

resultat = (x + y);
```

⚠ Les opérateurs d'affectation (=, -=, += ...), appliqués à des valeurs de type numérique, provoquent la conversion de leur opérande de droite dans le type de leur opérande de gauche.

```
// cela revient à faire (int)(x + y) car resultat est de type int
std::cout << "resultat = " << resultat << std::endl;
// Affiche : resultat = 6
```

Conversion automatique

Les conversions d'ajustement de type automatique réalisées suivant la hiérarchie ci-dessous sont réalisées sans perte :

char → short int → int → long → float → double → long double

unsigned int → unsigned long → float → double → long double

Nouveaux opérateurs de transtypage en C++

- **`static_cast`** : Opérateur de transtypage à tout faire. Ne permet pas de supprimer le caractère **`const`** ou **`volatile`**.
- **`const_cast`** : Opérateur spécialisé et limité au traitement des **`const`** et **`volatile`**
- **`dynamic_cast`** : Opérateur spécialisé et limité au traitement des ***downcast***.
- **`reinterpret_cast`** : Opérateur spécialisé dans le traitement des conversions de pointeurs peu portables.

La syntaxe est la suivante :

```
op_cast<expression type>(expression à transtyper);
```

où op prend l'une des valeurs (**`static, const, dynamic ou reinterpret`**)

```
char c = 5;
int a = static_cast<int>(c);
```

Pour l'héritage :

- transtypage « ascendant » (***upcast***) : changer un type vers son type de base (ne pose pas de problème)
- transtypage « descendant » (***downcast***) : conversion d'un pointeur sur un objet d'une classe générale vers un objet d'une classe spécialisée (dangereux).