

Allocation dynamique de la mémoire, les opérateurs new et delete

L'allocation dynamique :

L'opérateur new :

L'opérateur new permet d'allouer de l'espace mémoire dynamiquement ; c'est-à-dire, au moment de l'exécution du programme en fonction des besoins. Sa syntaxe est la suivante :

```
nom_pointeur = new type[taille];
```

L'exemple suivant montre l'allocation d'un espace mémoire pour 10 variables de type int

```
int* p;  
p = new int[10]; /* alloue de la mémoire pour 10 variables int et initialise le  
pointeur p avec le début du bloc mémoire */
```

On alloue de l'espace mémoire pour 10 entiers et on récupère l'adresse de cet espace dans un pointeur de même type.

- Il est possible de faire l'allocation directement au moment de la déclaration du pointeur.
- Si l'allocation mémoire échoue faute d'espace, la valeur du pointeur sera égale à **NULL** (NULL est une constante prédéfinie du C++ qui vaut zéro).
- Il est conseillé de vérifier que le pointeur n'est pas égal à NULL après l'utilisation de new.

L'exemple suivant montre l'allocation mémoire au moment de la déclaration et le test si l'allocation mémoire est réussie.

```
int* p = new int[10];  
if (p == NULL) cout << "L'allocation mémoire a échoué ";
```

L'allocation dynamique de la mémoire permet de définir des tableaux dynamiques dont la dimension est spécifiée au moment de l'exécution comme le montre l'exemple suivant.

```
int n;  
cout << " Nombre d'éléments ";  
cin >> n;  
int* p = new int[n];
```

L'opérateur delete :

L'opérateur delete permet de libérer l'espace mémoire alloué par new.

```
delete p;  
p = 0;
```

Même si on appelle l'opérateur delete sur un pointeur, celui-ci continue d'avoir la valeur de l'adresse de ce espace mémoire. Après l'appel de delete sur un pointeur, il faut le remettre à 0.