

locale automatique

Locales et attributs de classe, existent pendant exécution du bloc dans lequel elles sont déclarées/l'existence de l'objet.

Mémoire allouée en entrant dans bloc où elle est déclarée.

Construite au passage là où elle est déclarée.

Détruite et mémoire libérée en sortant du bloc où elle est déclarée (accolade "}").

emplace_back

Construit et insère un élément à la fin.

statique

Globales et locales statiques, début à fin du programme.

Mémoire allouée début programme.

Construite au 1er passage là où elle est déclarée.

Détruite et mémoire libérée après sortie de main().

push_back

Insère l'élément à la fin en le copiant. Si la capacité est dépassée, on la double (+1) (pareil avec emplace_back).

globale

Allouée et construite avant main().

Détruite et libérée après main().

allocation dynamique

Créées et détruites explicitement par programmeur.

Mémoire allouée et variable construite au passage sur l'instruction "new".

Détruite et mémoire libérée au passage sur l'instruction "delete".

Eventuellement dans une autre fonction.

allocation dynamique

On utilise new/delete lorsque l'on veut allouer de la mémoire et construire l'objet en même temps. Si l'on veut séparer ces 2 étapes, on utilise ::operator new(taille en bytes) et new(pointeur) objet, ainsi que pointeur->destructeur et ::operator delete(pointeur). On peut éventuellement caster le pointeur dans le type que l'on souhaite durant l'appel.