**« Résumé chapitre5 »**

‘‘ CONSTRUCTION ET DESTRUCTION ET INITIALISATION DES OBJETS ’’

Réalisé par :

ELYOUSFI Mohamed (BDCC1)

Supervisé par :

M. Khalifa MANSOURI

*Année universitaire 2020/2021*

1. **CONSTRUCTION ET DESTRUCTION DES OBJETS AUTOMATIQUES**

Les objets automatiques sont ceux crées par une déclaration :

* Dans une fonction, il est détruit à la fin de l’exécution de la fonction.
* Dans un bloc, il est détruit lors de la sortie du bloc.

En ce qui concerne la chronologie :

* Le constructeur est appelé après la création de l’objet
* Le destructeur est appelé avant la destruction de l’objet

1. **CONSTRUCTION ET DESTRUCTION DES OBJETS STATIQUES**

Les objets statiques sont ceux crées par une déclaration située

* En dehors de toute fonction.
* Dans une fonction, mais assortie du qualificatif **static.**

Les objets statiques sont crées avant le début de l’exécution de la fonction main() et ils sont détruits après la fin de son exécution.

1. **CONSTRUCTION ET DESTRUCTION DES OBJETS GLOBAUX**

Les objets globaux sont créés au début de l’exécution du programme et ils sont détruits après la fin de son exécution.

1. **CONSTRUCTION ET DESTRUCTION DES OBJETS TEMPORAIRES**

L’affectation d’un objet a un autre conduit a créé un objet temporaire (construction par recopie)

1. **CONSTRUCTION ET DESTRUCTION DES OBJETS DYNAMIQUES**

Déclaration : nomClasse \*objDyn ;

Allocation de mémoire : objDyn = new nomClasse ;

Utilisation : objDyn ->method ;

Suppression : delete objDyn ;

1. **INITIALISATION DES OBJETS**

Dans ce cas il faut utiliser le constructeur par recopie car :

* Il résoudre le problème de la même adresse pour deux pointeurs (si par exemple on a un tableau dans les attributs).

1. **ROLE DU CONTRUCTION LORSQU’UNE FONCTION RETOURNE UN OBJET**

* Lorsqu’un constructeur approprié existe, il est exécuté. S’il n’existe pas, aucune erreur n’est générée. Selon le contexte ceci nuira ou non au bon déroulement de programme.
* Il faut prévoir un constructeur par recopie lorsque l’objet contient une partie dynamique.

1. **LES TABLEAUX D’OBJETS**

Les tableaux d’objets se manipules comme les tableaux classiques.

Exemple : point courbe [50] ;

courbe[6].affiche() ;

courbe[8].symetrique(courbe[5]) ;

1. **OBJETS MEMBRES OU OBJETS D’OBJETS**

Exemple :

#include <iostream>

#include<conio.h>

using namespace std;

class point

{

int x,y;

public :

point(int abs=0,int ord=0)

{

x=abs;

y=ord;

cout<<"Constructeur point "<<x<<" "<<y<<"\n";

}

};

class poincol

{

point p;

int couleur;

public:

poincol(int,int,int);

};

poincol :: poincol(int abs,int ord,int coul) :p(abs,ord)

{

couleur =coul;

cout <<"Constructeur poincol "<<couleur<<"\n";

}

int main()

{

**poincol a(1,3,9);**

return 0;

}