**« Résumé chapitre 2 »**

‘‘ PROGRAMMATION ORIENTÉE OBJET : NOTION DE CLASSE ’’

Réalisé par :

ELYOUSFI Mohamed (BDCC1)

Supervisé par :

M. Khalifa MANSOURI

*Année universitaire 2020/2021*

1. **INTRODUCTION**

Le **C++** est un langage de programmation orienté objet, différemment à **C** qu’est un langage de programmation structurée.

Un langage orienté objet permet la manipulation de classes, qui contient des variables (données ou attributs) et des fonctions (méthodes) permet de manipuler ces variables.

1. **RAPPEL SUR LA NOTION DE PROTOTYPE DE FONCTION**

Le prototype d’une fonction est constitué du nom de la fontion, type de retour, type des paramètres.

typeRet nomFonction (typePara1,typePara2) ; //prototype réduit

typeRet nomFonction (typePara1 Para1,typePara2 Para2) ; //prototype complet

1. **LES STRUCTURES EN C++**

Déclaration : **struct** nomStruct {type var1 ;type var2 ;… ;} ;

Déclaration avec des fonctions membres :

**struct** nomStruct {

type var1 ;type var2 ;… ;

typeRet fct1(typePara1,typePara2,…) ;

typeRet fct2(typePara1,typePara2,…) ;

} ;

Définition d’une fonction membre :

typeRet nomStruct **::** fct1(typePara paramétre) {

traitements … ;

}

Utilisation :

Struct nomStruct a;

a.var1 = .. ;

a.var2=.. ;

//appel d’une fonction membre

a.fct1(x) ;

1. **NOTION DE CLASSE**

**Objet** : données dans lesquelles des méthodes sont appliqués

**Classe** : définit des données et méthodes d’un groupe des objets similaires.

Une classe est composé de trois niveaux de portée : private, protected ou public.

Les fonctions membres d’une classe sont définies de la façon suivante :

TypeRet nomClasse :: nomFct (paramétres){…}

**::** appelé l’opérateur de résolution de portée.

**Utlisation** :

nomClasse nomObjet ; //variable Objet / nomObjet.champ ;

nomClasse PoiteurObjet ; //Pointeur sur un objet / PointeurObjet->champ ;

**Notion de constructeur** : fonctions membre sans type, nommée par nom de la classe, exécuté **systématiquement** lors de la déclaration d’un objet.

**Notion de destructeur :** exécuté **systématiquement** lors de la fin de vie d’un objet, permet de prévoir toute action devant s’exécuter avant la destruction d’une instance, porte la forme ~nomClasse.

**Allocation dynamique :** lorsque les membres donnés d’une classe sont des pointeurs, le constructeur utilisé pour l’allocation dynamique de la mémoire et le destructeur pour libérer la mémoire allouée.

**Membres statiques :** pour permettre a tous objets de classe de partager les mêmes données.

Ex : class Exemple1 {

static int x ;

float y ;

} // x et le même pour tous les objets de la classe

**Veuillez trouver ci-joint avec ce document les exercices de tp1, ainsi que les différents exemples de cours.**