
Planche d'exercices N°2

Exercice 1 (références et pointeurs)

Ecrire un programme C++ qui :

- déclare un entier a;
- déclare une référence vers cet entier ref_a;
- déclare un pointeur vers cet entier p_a;
- affiche les variables, leurs adresses, la valeur pointée.

Exercice 2 (opérateur new)

Déclarer un pointeur vers un entier puis construire cet entier et l'initialiser avec une valeur entière. Afficher la valeur de l'entier et l'adresse de son emplacement mémoire.

Refaire la même chose avec un réel.

Exercice 3 (références et pointeurs)

Ecrire trois version d'une fonction qui teste l'égalité entre deux entiers (utiliser le passage par valeur, par adresse et par référence). Tester cette fonction dans main.

Exercice 4 (références et pointeurs)

Écrire une fonction nommée `incrémenter()` permettant d'incrémenter la valeur d'une variable passée en paramètre et une fonction nommée `permuter()` permettant d'échanger les contenus de 2 variables de type `int` fournies en argument :

- en transmettant l'adresse des variables concernées (seule méthode utilisable en C) ;
- en utilisant la transmission par référence.

Dans les deux cas, écrire un programme (`main`) qui teste les deux fonctions.

Exercice 5 (classe, constructeurs, méthodes, attributs)

L'objectif de cet exercice est de gérer les notes des étudiants d'une institution à l'aide d'une classe C++ Etudiant définie par :

Les attributs suivants :

- **matricule**: l'identifiant de l'étudiant (auto incrémenté)
- **nom**: nom d'un étudiant
- **nbrNotes**: le nombre de notes de l'étudiant
- ***tabNotes**: tableau contenant les notes d'un étudiant (allocation dynamique).

Les méthodes suivantes :

- Un constructeur d'initialisation
- Un constructeur avec arguments
- Un destructeur *~Etudiant ()*
- Un constructeur de copie *Etudiant (const Etudiant &)*
- Les getters et setters
- **void saisie ()** : permettant la saisie des notes d'un étudiant
- **void affichage ()** : permettant l'affichage des informations d'un étudiant
- **float moyenne ()** : retourne comme résultat la moyenne des notes de l'étudiant.
- **bool admis ()** : retourne comme résultat la valeur true, si un étudiant est admis et la valeur false, sinon. Un étudiant est considéré comme étant admis lorsque la moyenne de ses notes est supérieure ou égale à 10.
- **bool comparer()**: qui compare la moyenne des deux étudiants, retourne comme résultat la valeur true, si deux étudiants ont la même moyenne et la valeur false, sinon.

Exercice 6

(encore des classes, auto-référence, surcharge d'une fonction membre)

Réaliser une classe C++ "vecteur3d" permettant de manipuler des vecteurs à 3 composantes (de type float). On y prévoira :

- un constructeur, avec des valeurs par défaut (0),
- une fonction d'affichage des 3 composantes du vecteur, sous la forme : (x, y, z)
- une fonction permettant d'obtenir la somme de 2 vecteurs ;
- une fonction permettant d'obtenir le produit scalaire de 2 vecteurs.
- une fonction coincide permettant de savoir si 2 vecteurs ont mêmes composantes.

- une fonction qui renvoie la norme du vecteur
- une fonction nommée normmax permettant d'obtenir, parmi deux vecteurs, celui qui a la plus grande norme. On prévoira trois situations :
 - ◆ le résultat est renvoyé par valeur ;
 - ◆ le résultat est renvoyé par adresse, l'argument étant également transmis par adresse.
 - ◆ le résultat est renvoyé par référence, l'argument étant également transmis par référence.

Ecrire un programme utilisant la classe réalisée.

Exercice 7 (classe et fonction amies)

Créer une classe Point en C++ permettant de représenter des points dans un plan caractérisé par deux coordonnées x et y.

Définir les fonctions membres suivantes :

- constructeur d'initialisation et de copie
- affiche: se contentant d'afficher les coordonnées cartésiennes du point.

Définir les fonctions amies suivantes :

- comparer: vérifie si deux points sont identiques.
- distance: calcule la distance entre deux points.
- milieu: renvoie le point milieu du segment composé de deux points.

Exercice 8 (surcharge d'opérateurs)

L'objectif de cet exercice est de définir les opérateurs arithmétiques en C++ d'une classe Complexe en utilisant les fonctions membres et les fonctions amies.

Soit la classe complexe pour gérer les nombres complexes:

Attributs:

- **Re** la partie réelle de type double
- **Img** la partie imaginaire de type double

Méthodes:

- Constructeur avec arguments (valeurs d'initialisation par default)
- Fonction d'affichage
- Fonction qui retourne le module sachant que : $|z| = \sqrt{a^2 + b^2}$.
- Fonction qui retourne le conjugué sachant que : $\bar{z} = a - bi$

Surcharge des opérateurs suivants :

"+" : (complexe + complexe) sachant que : $(a + bi) + (c + di) = (a + c) + (b + d)i$;

"+" pour (complexe + double) et "+" pour (double + complexe)

De même pour "*", "-" sachant que : $(a + bi)(c + di) = (ac - bd) + (ad + bc)i$.

Créer un programme de test main ().

Exercice 9 (Associer entre les classes)

Ecrire une classe en C++ permettant de représenter des cercles qui se caractérisent par un rayon et un centre de type point (cf. exercice Classe Point)

Les opérations possibles sur un cercle sont :

- l'affichage des caractéristiques du cercle.
- l'obtention de son rayon,
- le changement de son rayon,
- l'obtention de son centre,
- la translation de son centre,
- le calcul de sa surface,
- le calcul de son périmètre,
- le test de l'égalité du cercle avec un autre cercle,
- la vérification de l'appartenance d'un point au cercle,

Prévoir deux constructeurs :

- Constructeur avec deux paramètres correspondant au rayon et au point centre
- Constructeur avec deux paramètres correspondant au rayon et aux coordonnées du point centre

Faire les modifications nécessaires à la Classe Point

Pour tester la classe, écrire un programme principal effectuant les opérations suivantes :

- création d'un cercle de rayon 1 et de centre (10,10) et affichage de ses caractéristiques
- calcul et affichage de son périmètre et de sa surface
- doublement de son rayon
- à nouveau calcul et affichage de son périmètre et de sa surface
- déplacement du cercle pour ramener son centre à l'origine
- vérification que le point (1,1) est bien à l'intérieur du cercle
- vérification que le point (3,3) est bien à l'extérieur du cercle
- test de l'égalité du cercle avec un autre cercle de centre (0,0) et de rayon 2