

## Laboratoire 7 : Révision

Fonctions, résolution de problème, implémentation de classe

- On veut analyser le niveau de réussite d'étudiants ayant passé un examen. Il s'agit de compter ceux qui l'ont réussi et ceux qu'ils l'ont échoué. Écrire un programme complet en C++ qui permet saisir les résultats avec 0 pour les étudiants qui ont échoué et 1 pour ceux qui ont réussi, puis affiche le décompte pour chaque catégorie. Si 80% des étudiants l'ont réussi, le programme affiche « Relever le niveau de l'examen ». Le programme permet de traiter les résultats de 10 étudiants.

- Écrire un programme C++ qui génère un nom d'utilisateur de la façon suivante : il demande le nom, le prénom et la date de naissance de l'usager, et concatène les deux premières lettres du prénom, les trois premières du nom de famille, le jour et le mois de la date de naissance.

### 3. Fonction et tableau

Écrire un programme C++ qui génère plusieurs noms d'utilisateur de la façon suivante : il demande le nom et le prénom de l'usager, et concatène les initiales au numéro séquentiel correspondant à ces initiales. Par exemple, le nom d'utilisateur de Marc Potvin sera MP1, celui de Michèle Poitras, qui s'est inscrite en second, sera MP2, et celui de Stéphane Kirouac sera SK1. Le programme doit aussi conserver une liste des noms d'utilisateur, afin de pouvoir l'afficher sur demande, et être en mesure de fournir le nom et le prénom d'un utilisateur, étant donné son nom d'utilisateur.

### 4. Débordement

La fonction *factorielle* est fréquemment utilisée dans des problèmes de probabilité. La factorielle d'un entier positif  $n$  (écrite  $n!$  et prononcée «factorielle  $n$ ») est égale au produit des entiers positifs de 1 à  $n$ . Écrivez un programme qui évalue les factoriels des entiers de 1 à 5. Affichez les résultats sous la forme d'un tableau. Quelle difficulté pourrait vous empêcher de calculer la factorielle de 20?

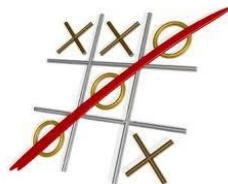
### 5. Jeu TicTacToe

Créez une classe **TicTacToe** qui vous permettra d'écrire un programme complet pour jouer au jeu de [tic-tac-toe](#).

La classe contient, comme données **private**, un tableau à deux dimensions d'entiers de 3 lignes et 3 colonnes.

Le constructeur doit initialiser le tableau vide avec des zéros.

Le programme doit être conçu pour permettre à deux personnes de jouer. Peu importe où le premier joueur se positionne, placez un 1 dans la case correspondante; placez un 2 à l'endroit où le second joueur se positionne. Chaque nouveau positionnement doit s'effectuer dans un carré vide. Après chaque positionnement, déterminez s'il y a vainqueur ou s'il s'agit d'une partie nulle.



## 6. Classe, tableau, constructeur, mutateur, accesseurs

Réaliser une classe nommée « Polynôme » représentant un polynôme de la forme  $a_0 + a_1x + a_2x^2 + a_3x^3 + \dots + a_nx^n$  avec  $n < 10$

et renfermant les fonctionnalités suivantes :

- un constructeur permettant de créer un polynôme nul de degré n,
- l'insertion du  $i^{\text{ème}}$  terme dont le coefficient est passé en paramètre,
- l'accès au coefficient du  $i^{\text{ème}}$  terme,
- l'évaluation du polynôme pour une valeur de x,

Réalisez également un programme permettant d'utiliser (pour démontrer qu'elles fonctionnent) chacune des méthodes développées.

## 7. Classe, surcharge des opérateurs

Créez une classe appelée **Complexe** capable d'effectuer des opérations arithmétiques avec des nombres complexes. Écrivez ensuite un programme pilote pour tester cette classe.

Rappel: Les nombres complexes possèdent la forme suivante:

$$\text{partie réelle} + i * \text{partie imaginaire} \text{ où } i^2 = -1$$

Utilisez des variables *double* pour représenter les attributs de la classe. Fournissez un constructeur, qui permet l'initialisation d'un objet de cette classe lors de sa déclaration. Le constructeur doit contenir des valeurs par défaut, dans le cas où aucun initialiseur ne serait fourni. Fournissez, des fonctions membres *public* pour chacune des opérations suivantes:

- Addition de deux nombres **Complexe**: les parties réelles sont additionnées ensemble et les parties imaginaires sont additionnées ensemble.
- Soustraction de deux nombres **Complexe** : la partie réelle de l'opérande de droite est soustraite de la partie réelle de l'opérande de gauche, tandis que la partie imaginaire de l'opérande de droite est soustraite de la partie imaginaire de l'opérande de gauche.
- Affichage des nombres **Complexe** selon la forme  $(a, b)$  où a représente la partie réelle et où b représente la partie imaginaire.
- Multiplication de deux nombres **Complexe** :  
si  $x=a+ib$  et  $y=c+id$  alors  $x \text{ fois } y = (a*c - b*d) + i(a*d + b*c)$
- Complexe conjugué d'un nombre **Complexe** : le signe de la partie imaginaire est inversé

