UNIVERSITÉ LAVAL
Faculté des sciences et de génie
Département d'informatique et de génie logiciel

Labo 12

Templates, STL, révision.

Questions de révision

1) Répondez à chacune des questions suivantes par vrai ou faux. Pour celles auxquelles vous répondez faux, expliquez pourquoi.
   a) Une fonction amie d'un modèle de fonction doit être une fonction de modèle.
   b) Si plusieurs classes de modèle sont générées à partir d'un modèle de classe unique avec un membre de donnée **static** unique, chacune des classes de modèle partagent une seule copie du membre de donnée **static** du modèle de classe.
   c) Une fonction de modèle peut être surchargée par une autre fonction de modèle portant le même nom de fonction.
   d) Le nom d'un paramètre formel peut être utilisé une seule fois dans la liste de paramètres formels de la définition du modèle. Les noms des paramètres formels parmi les définitions de modèles doivent être uniques.
   e) Les mots-clés **class et typename,** lorsqu'utilisés avec un paramètre de type de modèle, signifient spécifiquement « toute classe définie par l'utilisateur ».

2) Complétez les phrases suivantes:
   a) Les modèles permettent de spécifier, en un seul segment de code, une série complète de fonctions liées, appelées _____, ou une série complète de classes liées, appelées _____
   b) Toutes les définitions de modèles de fonction débutent par le mot-clé _____, suivi d'une liste de paramètres formels du modèle de fonction entourés par des _____
   c) Les fonctions liées, générées au départ d'un modèle de fonction, ont toutes le même nom, de sorte que le compilateur utilise la résolution de _____ pour invoquer la fonction adéquate.
   d) Les modèles de classe sont également appelés types _____
   e) L'opérateur _____ est utilisé avec un nom de classe de modèle pour lier chaque définition de fonction membre à la portée du modèle de classe.
   f) Comme pour les membres de donnée **static** des classes qui ne sont pas bâties sur un modèle, les membres de donnée **static** des classes de modèle doivent être initialisés sous la portée du ____
3) (Vrai ou faux) La STL fait un usage abondant de l'héritage et des fonctions virtuelles.
4) Les deux types de conteneurs de la STL sont les conteneurs de séquence et les conteneurs ____
5) Les cinq principaux types d'itérateurs sont: _____, _____, _____, _____ et _____
6) (Vrai ou faux) Le pointeur est une forme généralisée d'itérateur.
7) (Vrai ou faux) Les algorithmes de la STL peuvent opérer sur des tableaux pilotés par des pointeurs, comme ceux du C.
8) (Vrai ou faux) Les algorithmes de la STL sont encapsulés sous la forme de fonctions membres dans chaque classe de conteneur.
9) (Vrai ou faux) L'algorithme **remove** ne réduit pas la taille du **vector** dont les éléments sont en cours de retrait.
10) Les trois adaptateurs de conteneur de la STL sont: _____, _____ et _____

11) (Vrai ou faux) La fonction membre de conteneur **end()** donne comme résultat la position du dernier élément du conteneur.
12) Les algorithmes de la STL opèrent indirectement sur les éléments de conteneur par l'intermédiaire des _____
13) L'algorithme **sort** requiert un itérateur _____

Exercices

1) Expliquez pourquoi vous pourriez utiliser l'instruction suivante dans un programme en C++:
   Tableau< Employe> listeOuvrier(100);
2) Révisez votre réponse à l'exercice précédent. Maintenant, pourquoi pourriez-vous utiliser l'instruction suivante dans un programme en C++?
   Tableau< Employe> listeOuvrier;
3) Expliquez l'usage de la notation suivante dans un programme-en C++:
   template< class T > Tableau< T >::Tableau(int s)

4) Utilisez un conteneur de la STL pour implanter un mini-dictionnaire anglais-français. Ce dictionnaire devra fournir la traduction française d'un mot anglais fourni par l'utilisateur. Idéalement, ce programme devrait charger le dictionnaire à partir d'un fichier, mais vous pouvez aussi le charger directement en ajoutant les mots un par un dans le conteneur (mots en anglais et leur traduction française). Notez qu'un même mot anglais peut avoir plusieurs traductions en français

Exercices de révision

Soit cercle, carré, cube et sphere des formes respectivement à deux et trois dimensions. Développez une hiérarchie de classe pour représenter celles-ci. Cette hiérarchie doit comprendre la classe de base `Forme` à partir de laquelle sont dérivées les classes `FormeDeuxDimensions` et `FormeTroisDimensions`. Une fois votre hiérarchie développée, définissez chacune des classes `Cercle, Carre, Cube` et `Sphere` qui la compose. La classe de base `Forme` est une classe abstraite contenant l'interface vers la hiérarchie. Les classes `FormeDeuxDimensions` et `FormeTroisDimensions` doivent également être abstraites. Utilisez une fonction `affichage` virtuelle pour produire la sortie du type et des dimensions de chaque classe concrète. Ajoutez également des fonctions virtuelles appelées `aire` et `volume` afin que les calculs s'effectuent pour des objets de chaque classe concrète dans la hiérarchie.
Imaginez des contraintes pour fiabiliser le code et implantez la théorie du contrat.
Mettre en place le test unitaire pour chacune des classes pendant leur développement.

Réponses aux Questions de révision

1) a) Faux. Ce **pourrait** être une fonction de modèle; on peut aussi avoir des fonctions non modèles
2) . b) Faux. Chaque classe de modèle aura une copie du membre de donnée **static**. c) Vrai. d) Faux. Les noms des paramètres formels ne doivent pas être nécessairement uniques d'une fonction de modèle à une autre. e) Faux. Les mots-clés **class** et **typename** dans ce contexte permettent aussi l'usage d'un paramètre de type d'un type prédéfini (type de base comme int).
3) a) fonctions de modèles, classes de modèle. b) **template**, chevrons (< et >). c) surcharge. d) paramétrés. e) binaire de résolution de portée. f) fichier.
4) Faux. Ils ont été évités pour des raisons liées à leurs performances.
5) Associatif.
6) Entrée *(input)*, sortie *(output)*, aller *(forward)*, bidirectionnel *(bidirectional)*, à accès direct *(random access)*.
7) Faux. Il en est réellement un équivalent interchangeable.
8) Vrai.
9) Faux. Les algorithmes de la STL ne sont pas des fonctions membres. Ils opèrent indirectement sur les conteneurs par l'intermédiaire des itérateurs.
10) Vrai.
11) **stack, queue, priority_queue.**
12) Faux. En réalité, elle donne la position située juste après la fin du conteneur.
13) Itérateurs.
14) À accès direct.